ACCURATE JOINT DETECTION FROM DEPTH VIDEOS

TOWARDS POSE ANALYSIS

Longbo Kong

Dissertation Prepared for the Degree of

DOCTOR OF PHILOSOPHY

UNIVERSITY OF NORTH TEXAS

May 2018

APPROVED:

Xiaohui Yuan, Major Professor
Song Fu, Committee Member
Robert Renka, Committee Member
Hassan Takabi, Committee Member
Barrett Bryant, Chair of the Department of Computer Science and Engineering
Costas Tsatsoulis, Dean of the College of Engineering
Victor Prybutok, Dean of the Toulouse Graduate School

Kong, Longbo. *Accurate Joint Detection from Depth Videos towards Pose Analysis*. Doctor of Philosophy (Computer Science and Engineering), May 2018, 68 pp., 8 tables, 47 numbered references.

Joint detection is vital for characterizing human pose and serves as a foundation for a wide range of computer vision applications such as physical training, health care, entertainment. This dissertation proposed two methods to detect joints in the human body for pose analysis. The first method detects joints by combining body model and automatic feature points detection together. The human body model maps the detected extreme points to the corresponding body parts of the model and detects the position of implicit joints. The dominant joints are detected after implicit joints and extreme points are located by a shortest path based methods. The main contribution of this work is a hybrid framework to detect joints on the human body to achieve robustness to different body shapes or proportions, pose variations and occlusions. Another contribution of this work is the idea of using geodesic features of the human body to build a model for guiding the human pose detection and estimation. The second proposed method detects joints by segmenting human body into parts first and then detect joints by making the detection algorithm focusing on each limb. The advantage of applying body part segmentation first is that the body segmentation method narrows down the searching area for each joint so that the joint detection method can provide more stable and accurate results.

ACKNOWLEDGEMENTS

I would like to express the deepest appreciation to my major Professor Xiaohui Yuan, who has the attitude and the substance of a genius: he continually and convincingly conveyed a spirit of adventure in regard to research and scholarship, and an excitement in regard to teaching. Without his guidance and persistent help this dissertation would not have been possible.

I also would like to thank my committee members, Professor Song Fu, Professor Robert Renka and Professor Hassan Takabi, who gave me many helpful suggestions for my research. Their broad and rich knowledges in their research areas shown me many different possibilities and challenges in computer science. Without their tremendous support, this dissertation would also not have been possible.

Finally, I would like to thank all the members in CoVIS (Computer Vision and Intelligent System) lab, their gave me lots of help for my research. Especially, Amar Man Maharjan, who also works on human pose detection, gave me lots of great ideas about my research.

TABLE OF CONTENTS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. Overview

Human pose detection and tracking have been widely used in many fields in recent years, such as human-computer interaction, robot control, 3D animation creation and home entertainment. In 2010, Microsoft launched their first generation Kinect device, which is an image acquisition device that has multiple sensors, including a Time-of-Flight, camera. Since then, affordable 3D human pose detection device and technology becomes popular in our daily life, and application using 3D human pose detection embraces its boom. In a video-based detection and tracking system, the feature point of different body parts and joints such as head, elbows, and hands, provide the information of poses and activities of people. Many methods [26] [28] [27] [34] have been proposed to tracking feature points from RGB videos for human pose detection and tracking. When detecting human motions, joints provide sufficient information about the motion of a human subject. Hence, joint detection is vital for characterizing human pose and serves as a foundation for a wide range of computer vision applications such as physical training, health care, entertainment, etc [45], [32], [47], [41], [31]. For instance, knowing the precise location of human joints enables estimation of poses and movements, which facilitates personalized training for applications in rehabilitation and combat tactics instruction.

Many methods that use RGB videos to detect human poses have been proposed. However, such methods can be easily affected by lighting condition and shadows. Methods using RGB videos also suffer inaccurate 3D spatial information of the human body. Multi-view methods have been used for providing more accurate and richer spatial information of the human body, but still, suffer the same problem. In multi-view methods, multiple cameras are usually used to capture the data of the same object from different positions and view angles, and then all captured frames are combined together to provide relative complete information for the object. Strong priors are required to combine with optimization

procedure or inference steps to estimate human movements. Complex human motion detection from RGB videos still remains an open problem. Comparing with visual (RGB) and infrared cameras, depth cameras shows a great advantage by providing the three-dimensional information. Most of the current research choose Time-of-Flight (ToF) camera (one kind of depth camera). The time-of-flight camera measures the time between each signal pulse and receiving the reflection of the signal. Then the distance is calculated by multiply the speed of the light and the measured time. In a depth frame, each point carries the distance between the target and the camera. With the intrinsic parameters of a depth camera, the three-dimensional position of each point from the captured point cloud is acquired.

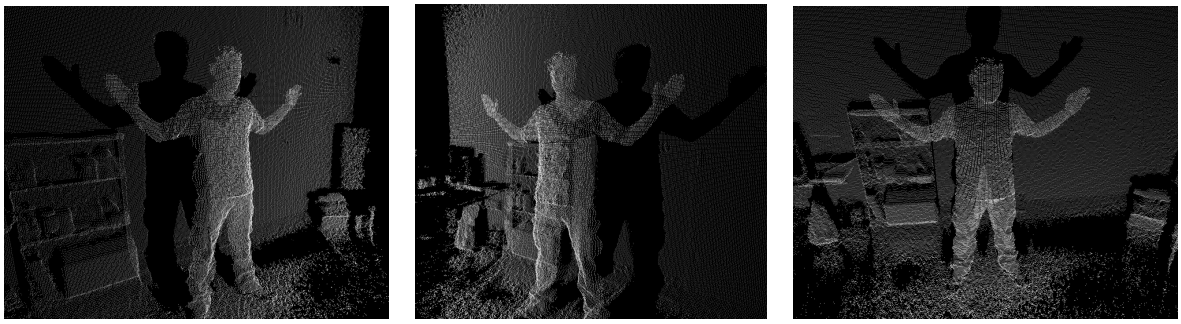


FIGURE 1.1. Examples of three different views for the point cloud in one frame captured by Microsoft Kinect.

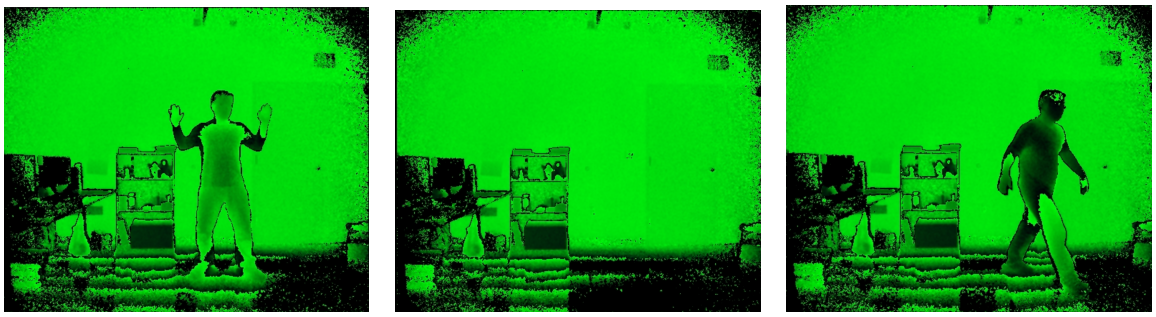


FIGURE 1.2. Examples of depth frame projected by point cloud captured by Microsoft Kinect. The distance value is mapped to the color channels of RGB image.

1.2. Applications

As mentioned in Section. 1.1, there are many scenarios and applications are using human pose detection technology, and more starts to embrace the technology. Microsoft brought human pose detection technology into the game industry first in 2010. In their system, users can simply play games by using natural actions instead of remote controllers. Such system extracts feature points from the human body as input for game applications. Studies about pose detection in gaming scenarios are also conducted. Ke et al. [18] proposed a real-time method to estimate 3D human poses from monocular camera automatically for event detection and video gaming scenarios. In their method, information of human bodies such as edge, color, and silhouette are extracted. Then, the extracted features are used as the input for game applications. Human pose detection also can be used in healthcare and medical fields. Fall detection is one of the most studied topics. Fall detection is an important application because computers can always monitor the activities of patients and notify the medical staff immediately when patient fall down. Fall detection system can help hospital or nursing home improve the quality of their service and save lives when dangers happen to patients. Most of the fall detection system detects the fall activity by extract the human pose first. Then the extracted data of human pose is analyzed to determine if the human subject fall down. In [4], Bian et al. proposed a fall detection method by tracking key feature points on the human body with a ToF camera. In their method, key points are extracted by applying a randomized decision tree. Ma et al. [23] also proposed fall detection method by using a ToF camera. In their work, a shape feature based method and a machine learning based method are combined to detect falling of human subjects. Human pose detection is also used in smart home systems. In a smart home system, one or multiple sensors are installed in the user's house to capture motion or voice command. Chun et al. [11] proposed a smart human motion based lighting control system. In their system, human actions and movement are detected and tracked over time, and the color temperature and illumination are changed based on the understanding of different human actions.

Smart surveillance is an application that can be dramatically improved by human

pose detection technology. Traditional surveillance system requires security staff monitor multiple screens and to enforce the safety of the public. However, human beings are not capable of monitoring complex data for a long time. Using human pose detection technology, computers can automatically extract pose information from the captured data and analyze the information. Based on the human pose detection technique, computers can recognize different activities through all kinds of poses. Thus, dangerous activities can be recognized immediately.

Human pose detection is also the foundation of many human activities recognition methods, such as the methods proposed in [33], [7], [3], [8]. Sung et al. [33] proposed a method for detecting and recognizing unstructured human activity in unstructured environments. In [33], the extracted human poses are used along with color information to recognized different activities. In [7], the authors proposed an evolutionary algorithm to detect human poses by selecting optimized feature points on the human skeleton model. The objective of their method is to improve the accuracy of human action recognition using RGB-D devices by only selecting the joints that are relevant to the corresponding activities. Bengalur et al. [3] proposed a method for human activity recognition using support vector machine (SVM) classifier. The extracted 3D human skeleton was used as a compact representation of human poses. Therefore, it is believed that accurate information of human poses can dramatically improve the performance of human activities recognition methods. In summary, human pose detection method can be widely used in many modern applications and research areas as well. Methods that provide more accurate detection results and better performance or aims to address unsolved problems will benefit more applications. More and more applications will start to embrace human pose detection technologies to provide a better experience.

1.3. Problem Statement

When detecting human poses in depth videos, there are three major strategies: model-based, learning based, and local data-driven methods. Model-based methods usually define their generic 3D human body model, then try to fit the model into the acquired point cloud. In model-based methods, all body parts and joints are defined by the model. Therefore, once

the model fitting procedure is done, the joints and other body parts are automatically located. However, the accuracy of joint detection is suboptimal due to misalignment, which affects the precision of tracking human movements. The predefined 3D model of human body also affects the overall performance of model-based methods. Because feature points are attached to the predefined 3D model, yet the shapes and proportions of human body vary from person to person. There have been many learning based methods proposed in recent years. Learning based methods mainly rely on trained classifiers to classify the points on the human body. Yet, the outliers in the results of classification affect the stability and accuracy of identifying the joints. Hence, the results of human poses are affected by inaccurate joints. The training dataset also affects the detection results. To achieve robust classification results, large scale dataset is required to cover as many cases as possible. Shotton et al. [31] used over 1 million images as their training dataset to train their classifier, to obtain robust classification results. One advantage of learning based method is robustness to the deformation of the clothes on human body. Another strategy is the data-driven method. Data-driven methods usually focus on the acquired data without knowledge from predefined model or training procedure. Data-driven methods try to detect human poses by analysis the features to detect human poses, such as shapes or geodesic relationship between feature points. However, data-driven methods cannot label the detected feature points as corresponding body parts. Because the acquired data from depth camera doesn't carry any body part information. Therefore, data-driven methods usually collaborate with other systems e.g. database, weak model or template, to label the detected feature points. The model-based and learning based methods usually require multi-threads computing or GPU acceleration to make the system able to run in real-time [32], [37], [31]. Because the optimization procedure and pixel-wise classification are heavy computing tasks to modern hardware. Different with model-based and learning based methods, data-driven methods are more scalable compared to the other two methods and require less computing capability. This advantage makes data-driven methods potentially can be deployed on smaller devices with weak computing performance, such as mobile devices and IoT (Internet of Things) devices.

In summary of those three major strategies discussed above, the human poses or motions are generally described by the feature points on the human body. Therefore, accurate detection of joints is necessary for detecting and tracking the motions and activities of a human and vital for delivering accurate results of detecting and tracking human motions. However, there are still many issues remain unsolved. Optimization during the model fitting can leads to local maxima. 3D human model with fixed size can directly cause inaccurate pose detection results because of the diversity of different human bodies. Ambiguous boundary caused by outliers also affect the accuracy of the joint detection. To overcome the aforementioned issues, we first proposed a hybrid framework, which integrates data-driven and model-based strategies, to take advantages of both strategies to provide stable and accurate joint detection results. Then, another joint detection based on body part segmentation is also proposed. In the second method, a data-driven body part segmentation method is proposed to make the joint detection have even less dependency on predefined models.

1.4. Overview of The Proposed Method

Despite recent developments in markerless human tracking, few methods, to our best knowledge, have been proposed for accurately detecting human joints. The objective of both proposed work is to accurately detect joints in the human body. The first method is to detect joints by combining body model and automatic feature points detection together. Fig. 1.3 represents the workflow of the first proposed framework. The human body model maps the detected extreme points to the corresponding body parts of the model and detects the position of implicit joints. The dominant joints are detected after implicit joints and extreme points are located by a shortest path based methods. The main contribution of this work is a hybrid framework to detect joints on the human body to achieve robustness to different body shapes or proportions, pose variations and occlusions. Another contribution of this work is the idea of using geodesic features of the human body to build a model for guiding the human pose detection and estimation. The second method detects joints by segmenting human body into parts first and then detect joints by making the detection algorithm focusing on each limb. The advantage of applying body part segmentation first

is that the body segmentation method narrows down the searching area for each joint so that the joint detection method can provide more stable and accurate results. The human body segmentation methods still use shortest paths as the input. The method follows the direction of each shortest path that starts from the same extreme point and determines the boundary of the body part by calculating the angles between different paths. It is assumed that the directions between different paths are not significantly different when they are in the same limb, on the other hand, the directions between different paths are dramatic when they go into various body parts. Fig. (1.4) illustrates the workflow of the proposed human body segmentation based joint detection method.

The rest of this dissertation is organized as follows: Chapter 2 reviews the related work on human pose detection. Chapter 3 presents the hybrid framework for detecting joints in the human body. Chapter 4 discusses the experimental result and compares the proposed method with other state-of-art methods. Chapter 5 is the conclusion and the future work.

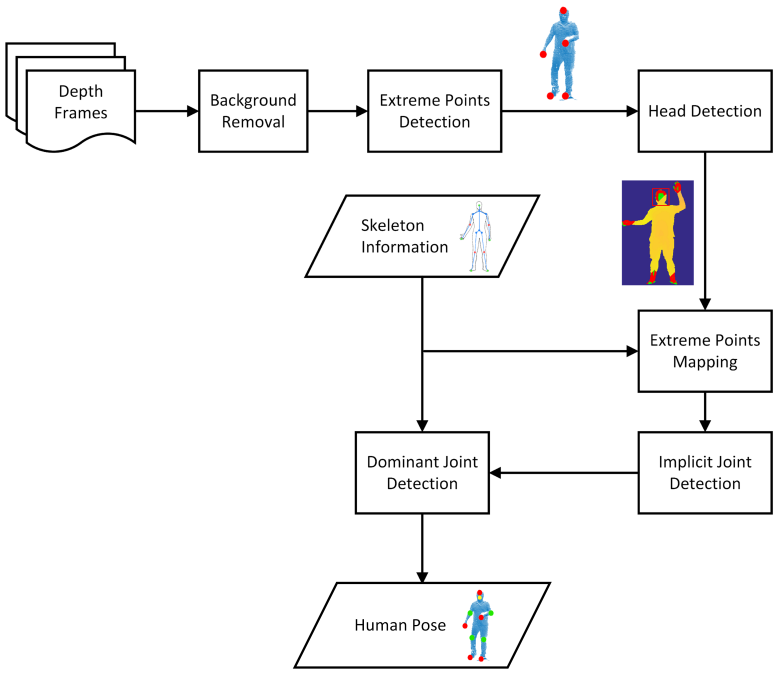


FIGURE 1.3. Overview of the workflow of the proposed hybrid based joint detection method.

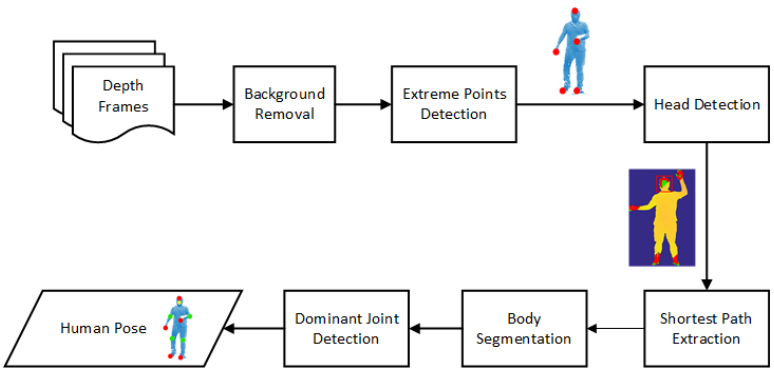


FIGURE 1.4. Overview of the workflow of the proposed segmentation based joint detection method.

1.5. Contribution

The first contribution of the HJD method is that a hybrid strategy, which combines both data-driven and model-based methods, is proposed. The data-driven and model-based methods work closely by providing information to each other in many steps of the proposed system. The model-based method in HJD takes full advantages of the natural features of human body and geodesic relationships between different body parts to detect weak joints. The second contribution of the HJD method is that the joints on the human body are categorized into two different types and detected with different methods.

The main contribution of the SJD method is that a joint detection method based on the results of human body segmentation is proposed. The human body segmentation method takes advantage of the natural features of the human body to divide the human body into limbs and torso. A data-driven strategy is used in the proposed human body segmentation method so that less prior knowledge about the human body is required.

CHAPTER 2

RELATED WORK

2.1. Data-Driven Methods

Markerless human body pose detection is an active research topic in computer vision area. Feature points detection, 3D human body model fitting or alignment, and pixel classification are the most widely used strategies. Methods that detect feature points usually use the silhouette of the human body as a graph and apply shortest distance algorithm to locate feature points. The human body is represented by the detected feature points. Plagemann et al. [29] proposed a real-time body parts detection and identification method using depth data. In their method, the human body is treated as a graph, and the points are treated as nodes on the graph. Interest points that represent different body parts are extracted by applying Dijkstra's algorithm on the graph of the human body. The adaption of the Dijkstra's algorithm is base on the insight that geodesic distances on a surface are largely invariant to the surface deformations or rigid transformations. The orientation of each interest point is calculated by back-tracking the path which leads to the interest point. The calculated orientation of interest points can be used for analysis the human poses and activities. Then, the author applied a body part identification method to classify detected feature points into different body parts. It extracts the shape of the area around each detected feature point and compared with shape features of different body parts from the trained data set. However, joints remained undetected in [29]. Baak et al. [2] proposed a data-driven hybrid strategy, which combines local estimation and global database lookup techniques, to estimate the human poses from depth frames in real-time. In [2], a point registration method was proposed to estimate human poses by using real-time captured data. The point cloud registration is based on an efficiently sparse Hausdorff distance. Feature points were detected by a modified Dijkstra's algorithm, then the extracted feature points were used as query input to search similar poses from the database. The final estimation result for the human pose is voted by both systems. In their method, hybrid strategy leverages the advantage of each sub-system

and combine them together to improve the overall performance. The local optimization method provides high frame rate but unstable results due to the noise of the captured data. The global optimization, which uses database lookup technique, provides stable results, but the runtime of global optimization is not suitable for real-time applications. Therefore, local and global optimization methods were combined together to deliver stable results at a real-time frame rate. Specifically, the database based global optimization is used to overcome the limitations of local optimization. To make the global optimization procedure able to work in their proposed hybrid framework, the author selects five endpoints on the human body as the feature points to reduce the complexity of the searching procedure. In [2], joints were defined on the human body model that was used for point cloud registration. Feature point detection method only detected geodesic extrema on the human body.

Li et al. [22] proposed a local shape context descriptor for describing the shape features of different body parts. The overall strategy in the proposed method is similar to the work proposed in [29]. In [22], extreme points on the human body were detected as well, then local shape descriptor was extracted from the edge images which are converted from the original depth images of the area around each extreme point. The proposed local shape descriptor was a set of distance value from a reference point to its nearest edges. The distance value was uniformly sampled in radial directions. Then, interest points are defined base on the extracted descriptors. The hierarchical searching algorithm is used to search interest point. To classify the detected extreme points into corresponding body parts, the author used a multivariate Gaussian Model to model the LSC (local shape context) features for each category. The mean and covariance matrix for each Gaussian model of each category is specified by using LSC features from manually selected frames with different shapes for each extreme point. Comparing with a deterministic template, the multivariate Gaussian model can handle the body parts with significant shape deformations, such as opening and closing a hand.

Handrich et al. [15] proposed a method for human pose estimation based on geodesic distance features from depth frames. The proposed method detects the pose of upper body

because in most of the human-computer interaction people only use their upper bodies to send commands to the computer. In [15], the author treated the human body as a graph and detect extreme points by measuring the pose-independent geodesic distance on the surface of a human body. Then, the detected extreme points are used to fit the pre-defined kinematic skeleton model into the point cloud. The landmarks of elbows are also detected in [15]. The landmarks of elbows are detected based on checking the curvature of the path between hand and shoulder. In the triangle formed by hand, shoulder, and elbow, the elbow is defined as the point that maximizes the area of the triangle on the path between shoulder and hand. However, the proposed elbow detection can be affected by the deformations of the cloth and body shapes of the human subject.

Jala et al. [17] proposed a hierarchical pose estimation method using ridge data from depth frames. The ridge information is extracted using both silhouette and depth data. The silhouette of the human body is used for determining the connectivity of neighboring points in the point cloud and setting boundaries for extracting ridge data. The depth data is used to calculate the ridge data by identifying the local maximal across each pair of edges and finding a chain of points as the ridge. Body parts are segmented to track over time. The torso and head are modeled with predefined templates, and the limbs are detected by applying Hough line detection. When detecting limbs, the ridge data is used to form straight lines by using Hough line detection, and then the junction point of two lines is considered as the joint on the corresponding limb. In [17], the tracking system helps to predict body parts when they are hidden or incorrectly detected. By using ridge data, less data is used for computing during the estimation and tracking. Therefore, multiple human subjects can be estimated and tracked in real-time without GPU laceration technique. Different with most of the graph based data-driven methods, the method proposed in [17] provided a different aspect for researchers to look at the depth data, and a new idea to estimate the human poses. Xiao et al. [39] proposed a 3D body parts segmentation method using topological and geodesic distance features of the human body. In [39], discrete Reeb graph (DRG) is used to represent the features of the human body. The Reeb graph is used to encode compact

manifolds in 3D. The Reeb graph gives one-dimensional representation for 3D objects. In the proposed method, three graphical patterns of DRG are defined in order to segment the human body into parts. The DRGs are extracted from the 3D scanned human body and then categorized into the defined patterns. Geodesic distance is used as the Morse function to construct the DRGs.

Data-driven strategy leverages local features, such as geodesic feature and shape feature, to extract information from human poses. It does not rely on any predefined model or prior knowledge. However, the data-driven method cannot map the detected extreme points to the real human body parts due to the lack of the knowledge of human body. In most of the cases where extreme points are detected, the point cloud of a human body is treated as a graph, and each node in the point cloud is treated as a node on the graph. No labels are assigned to any of the points on the human body. For example, the algorithm cannot determine if a detected extreme point belongs to hand or foot due to lack of definition of hand and foot. Therefore, a data-driven method is usually proposed with another subsystem to provide meaningful labels to the detected extreme points. Furthermore, shortest distance rule is only applicable to extreme points (endpoints of the human body). Joints are remained undetected or estimated by the predefined human body or skeleton model.

2.2. Model-Based Methods

Model-based methods are the most classic methods among all three major categories. Before depth camera become popular, many model-based methods [24], [25], have been developed to estimate human poses using monocular camera. There are many methods [45], [38], [32], [15], [30], [35], [44], [12] using human body model has been proposed to detect human poses with depth camera. Zhang et al. [44] proposed an approach to estimate human poses based on the data-driven Markov Chain Monte Carlo (DDMCMC) framework. Two efficient Markov Chain dynamics under the Markov Chain Monte Carlo framework is proposed to reduce the high dimensional state space into lower dimensional space. The two Markov Chain dynamics are diffusion and jump, the diffusion represents the local searching operation, and the jump operation indicates switching to a new local optimization procedure.

Then, A three-level tree structure based on the image observation and body topology was used as a human body model and the pose estimation is formulated as a Bayesian inference problem. The tree structure state space is parsed into an ordered set of body parts based on the topology of human body and observations. The tree structure provides a smooth and natural way to incorporate prior knowledge with optimization steps.

Cui et al. [12] proposed a method which integrated both low- and high-dimensional tracking approaches into a framework using a probabilistic fusion formulation. The low-dimensional was designed to overcome the high-dimensional problem of motion tracking; whereas the high-dimensional approach was designed to track movements by sampling the pose space without a trained model. On the other hand, the low-dimensional approach can only detect limited types of motions, and the high-dimensional approach lacks robustness and efficiency. In their proposed method, the low-dimensional approach requires training procedures to learn the motion models. The high-dimensional approach requires no training procedure. The low-dimensional and high-dimensional approaches work parallelly so that the overall performance was improved by concentrating on their advantages and resolving their weak points respectively. A set of probabilistic fusion criteria is used for fusing the two approaches. Both approaches try to remain their own states according to their confidence, and the one with higher confidence is more likely remain its own states. Then, the system selects the results from the approach with higher confidence. In addition, the low-dimensional and high-dimensional approaches complement each other by an updating scheme. In the updating procedure, one approach updates its own detection state with the state from another approach with a probability. As a result, the weaker approach is more likely updated by the approach with strong confidence.

Zuffi et al. [47] proposed a method that used realistic and part-based 3D human models for human pose detection. Different from traditional model-based methods, the authors defined their human body model by separating the human body into independent parts. Each body part was defined by two subspaces of shape deformations and mean shape. The subspaces of shape deformation were learned by using principal component analysis

(PCA). As a result, each body part can be translated, rotated and deformed independently in 3D space to fit the data in a wider range than traditional all-in-one models. The human body in [47] was represented by a graphical model whose nodes on the graph belong to different body parts. A cost function is defined to calculate how smoothly two adjacent parts can be connected. The pose and body shape were inferred by using particle-based max-product belief propagation. By using the particle-based max-product belief propagation, the part-based model is granted computational advantages due to the distributed structure. Furthermore, breaking human body into independent parts allows the parameters of each body part to be estimated independently from the data. However, it is difficult to recover body shape for such methods, due to the crude model. The all-in-one model-based methods use relatively high dimensional state space to fit their model into the captured data. As a result, such methods usually face challenges in computing. In [47], the author takes advantages of both distributed parts method and traditional all-in-one model base method by proposing their new stitched puppet (SP) model.

Ye et al [41] proposed a method with both pose estimation and pose refinement from a single depth image. To initialize the system, captured depth data was matched to a set of pre-captured motion template. Then the initialization step generates body configuration estimation and semantic labels on the human body base on the template. The pre-captured dataset contains 3D human body meshes, as well as the corresponding skeleton data. The pose refinement method is to correct the estimation results by fitting the estimated pose data represented by skeleton back to the input point cloud. The refinement procedure uses a non-rigid point cloud registration to calculate the difference between the template and the original point cloud and to fill the missing regions caused by occlusion. During the refinement process, the size of the database remains small compared with the method that directly maps template into the original point cloud. The initial estimation also prevents the refinement process from being trapped in local maxima. Ye et al [41] also proposed a view-independent matching algorithm to match their 3D full body surface template into the captured point cloud. Simply matching a 3D surface mesh with point cloud usually leads to

inaccurate estimation. In [41], the author applied principal component analysis (PCA) on both original point cloud and their template database. Then, both results after PCA are aligned in three principal axes. A template that matches the point cloud can be searched in a reduced space.

Recently, Sigalas et al. [32] proposed the top view reprojection method to estimate human poses in RGB-D videos by aligning body model to the point cloud of the human body. The proposed method provided a new way to leverage 3D human body, and a new point of view to look at the 3D point cloud. In [32], a cylinder based human model is used for representing human body. A set of hypotheses are generated for each body part and tracked by a particle filter. Then the points inside the cylinder body part model are re-projected to the top surface of the cylinder body part. The ratio of the number of re-projected points to the total number of points inside the cylinder model is computed as a re-projection ratio. The best hypothesis position of a body model is determined by selecting the minimum top view score (TVR) which includes the reprojection ratio, alignment term, and discrepancy term. The reprojection ratio serves as the key factor in the scoring function. The alignment term is responsible for adjusting the alignment between original point cloud and the hypothesis. The discrepancy term is used for favoring the hypotheses with large overlapping areas between the model and point cloud. The discrepancy term also penalizes invalid hypotheses. The joints were defined on fixed positions of each cylinder body part model. Methods using model fitting strategy usually have great complexity due to the optimization steps. Self-occlusion and occlusion caused by other object or human are also challenging to a model-based method. However, different human subjects can be separated by proposing TVR (top view reprojection) framework. Self-occlusion is also handled by the scoring function proposed in [32]. The scoring function is applied to each body part independently. Therefore, all body parts are estimated uniformly and independently without intermediate descriptor of body parts. The TVR framework also has high tolerant to different human shapes, by adopting the scoring function.

The proposed method in [15] detects the pose of upper body by proposing a hybrid

framework combining data-driven and model-based methods. In [15], the data-driven method is responsible for detecting landmarks of body parts. Then, the detected extreme points are used to fit the pre-defined kinematic skeleton model into the point cloud. The skeleton fitting method is responsible for fitting the skeleton model into the point cloud based on the detected landmarks, and reject incorrectly detected landmarks. Inverse kinematics is used to align the skeleton model with the detected landmarks. Due to the detection of landmarks, the skeleton model can fit into point cloud without fix the size of each bone segments. As a result, the method in [15] can fit a wider range of body shapes and less person-dependent. Schwarz et al. [30] proposed a hybrid strategy, which contains both data-driven method and model-based method, to estimate human poses. Here, it is categorized as a model-based method, because the data-driven method in [30] only plays a supporting role in the whole system. In [30], Schwarz et al. applied Dijkstra's algorithm to detect endpoints of limbs as primary feature points. Then, a skeleton model was fit into the point cloud using constrained inverse kinematics. Graph-based landmark detection provides more stable results comparing with methods that rely on appearance-based features for landmarks detection. For the poses that contain body part occlusions, motion information provided by optical flow between depth frames is employed to resolve the occlusion issue. A body segment map is generated to indicate the location of the entire occluding body segment in the depth frame. This body segment map is updated over time. Joints were defined by the skeleton model, and the positions of joints were estimated by the model fitting procedure.

Zhu et al. [45] proposed their human pose estimation approach using model-based strategy and Cartesian control theory. Similar to the work presented in [30], features that represent positions of anatomical landmarks are extracted from depth data and tracked over time. Then, the extracted features (anatomical landmarks) are feed into a constrained, closed loop tracking algorithm to estimate the pose of the articulated body model. In addition, the tracking algorithm also provides feedback to the feature extraction method to address ambiguous features and estimate the features that are failed to be extracted. The author also proposed a tracking framework to enforce constraints on joints and an avoidance of

self-penetration. In their proposed framework, feature extraction dramatically reduces the complexity of the model fitting by simplifying a large number of degrees of freedom into a small number of features. Yet, all joints are defined as the intersections of two connected body part templates. Ganapathi et al. [13] proposed a filtering algorithm for tracking human pose from depth videos by combining an accurate generative model with a discriminative model. In their system, a local model-based search, which takes advantages of the features of kinematic chain, is applied in each iteration filter. The discriminant model is a set of trained patch classifiers, which are used to provide information of body part locations when the model-based local search, is disrupted by fast movements or occlusions. In addition, the proposed discriminant model also reinitializes the model-based local search when the local search fails.

It is clear that model-based or model involved methods usually define their human models by defining the body parts and degrees of freedom of corresponding parts, or kinematic relationship between connected body parts. Such methods usually provide stable and smooth results for human pose estimation, especially comparing with data-driven methods. However, these methods suffer from different body proportions, cumulative error of model fitting and local maxima during the model fitting. Moreover, a fixed proportion of the body parts on the models can lead to ambiguous or even inaccurate joint locations. The demand for computing caused by optimization process also limits the application of such methods, especially in applications that can only provide weak hardware such as cellphone or IoT devices.

2.3. Learning Based Methods

For human pose estimation problem, using machine learning allows the proposed system to classify body parts and estimate human poses without relying on a predefined model at runtime. Model independence allows the system to avoid many issues in model-based methods, such as local maxima during the model fitting procedure and inaccurate results caused by inaccurate or over fitted human body model. Similar to the model-based methods, most learning based methods also focus on large body parts detection. Because

comparing to joints, body parts have a larger area of the human body. The Larger area of the human body means more points on the point cloud to learn and classify, therefore, more accurate results can be generated. In learning based methods, joints are also usually defined as the intersections of connected body parts.

Wei et al. [37] applied classification method in their system to detect the initial pose and registered the human skeleton model to the depth frame. After the skeleton model is initialized, a tracking method is invoked to track 3D poses via a Maximum A Posteriori (MAP) framework. Randomized trees are used to label the point on human body automatically. Randomized trees are suitable for multi-class classification and can provide robust results. After the human body is segmented by labeling, the skeleton model is registered to the point cloud based on the segmentation result. During skeleton registration procedure, the size of each bone segment is modified to fit the segmentation result. Therefore, the skeleton model is suitable for different body shapes and proportions. The tracking system relies on a real-time point cloud registration method in MAP framework. Within this framework, there are four terms: depth image term, extra depth term, silhouette image term and prior term. The depth term evaluates the likelihood of hypothesis and the original data. The extra depth data is used to fix the incorrect registration result from depth image term due to camera noise or significant occlusion. The silhouette term is used to penalize the mismatch result between registered pose and original data. Finally, the prior term is used to measure how smooth the current result is placed on previous results. The joints were defined by the skeleton model and initialized by the classification method at the beginning. The work in [37] combines both human pose detection and tracking methods, and take advantages of both methods. The learning based method segments the human body so that the skeleton model can be calibrated and registered into the human body. The learning based method also recover the system when the tracking method fails to track the human poses.

Shotton et al. [31] proposed two algorithms to estimate human poses, body part classification (BPC) and offset joint regression (OJR). Both algorithms only rely on the machine learning technologies to ensure they can work without manually initialization or calibration

for any human body shape. Both algorithms adopt efficient decision forest to evaluate the contribution of every point to each joint. The BPC algorithm employs intermediate representations of body parts, in order to achieve an accurate classification of body parts in pixel level. Then, the BPC algorithm provides proposals for joints on the human body with weighted confidence. The OJR algorithm regresses the positions of the joints on human body directly instead of segmenting the human body into parts first. The difference between BPC and OJR is that they use different labeling method for training data and different leaf node prediction models in their randomized forest. The labels on the training data for BPC includes 31 body parts, each limb is divided into upper and lower parts. For OJR, only 16 body parts are defined. The leaf node prediction model in BPC predicts a label for each point. Prediction for each point is used as an intermediate step for predicting joint positions. On the other hand, the leaf node prediction in OJR directly predicts a set of weighted votes for each body part. In [31], the authors provided a complete methodology for human pose estimation with machine learning and a new way to select features from a human body for machine learning algorithm.

Buys et al. [5] proposed a customizable and adaptable system for human pose estimation and tracking using the RGB-D camera. The proposed system can be applied to different applications and can be used for the cases in which the depth camera is moving instead of one fixed position. In [5], each pixel in the depth frames is considered as a body part candidate. Multiple skeleton data is extracted from all the hypotheses. Furthermore, their method also adopts an appearance model which combines depth value with color information. The appearance model is to label the pixels which belong to the human subject in depth frames. During the runtime, every pixel in each depth frame is classified as a body part. Then, body part proposal method is applied to refine the rough classification result from the previous step. From there, kinematic tree searching is applied to the proposals of body parts. At this step, the result is still noisy and not complete. Then, the pixel classification, body part proposal, and kinematic tree searching are applied one more time to generate a new estimation result. To obtain a new estimation result, the appearance model

is used in the pixel classification part. The estimated human pose from the first iteration is then used for retrieving missing body part and refine the existing part.

Abobakr et al. [1] proposed a method for joint detection by formulating pose estimation task as an offset joint regression problem and using deep convolutional neural networks to locate the joints from depth frames. The deep convolutional neural networks try to learn representations of a human body from depth frames in multiple levels. This learning procedure is composed of simple but non-linear modules, each module converts the representation from its current level to a higher and more abstract level. The advantage of deep learning methods is that the representations of a subject, such as human body, can be trained, therefore, handcrafted features are not highly demanded. However, to ensure the deep learning model fully functional, a large amount of training data is needed. To address this problem, the author also modified the state-of-the-art synthetic data generation system to generate high quality training dataset in which each frame provides the position of joints on a human body. Li et al. [21] proposed learning method based on deep neural networks for pose estimation. The author focuses on joint detection on the human body and converts human pose estimation problem into structured-output task based on the dependencies among all the joints on the human body. In their method, the captured image and 3D human model are used as input, then the likelihood of the captured data and 3D human model is calculated. The network in [21] contains a convolutional neural network for extracting features from frames and two sub-networks. The sub-networks transform the extracted features from frames and the input 3D pose into a joint embedding. The likelihood function and image-to-pose embedding are jointly trained with a maximum-margin cost function.

Chandra et al. [9] proposed human body segmentation method using deep learning algorithm. The objective of the proposed method is to estimate human poses from the segmented body parts. In their work, a manual annotation scheme for videos is developed so that annotating segmentation masks in each frame is no longer required. The author also extends the state of the art deep learning system to use both color and depth information in one framework. The annotation scheme is developed based on a clustering-based method

which reduces the whole dataset into a set of representative frames. In the clustering method, the histogram of oriented gradients (HOG) features is calculated across all frames. Then, Euclidean distance between adjacent frames is calculated. At last, frames with Euclidean distance less than a fixed threshold are clustered into the same group. The represented frame is randomly selected from each group. Once the represented frame is labeled, the result will pass to the remaining frames in the same group. The deep learning framework in [9] is built based on the Deeplab network [19]. To make the Deeplab network work with depth data, the author encodes the depth value in each pixel into a 3×1 vector. The vector represents the height above the ground and the angle between the normal of the surface and the gravity direction. Then the value is scaled to the range from 0 to 255. In [9], Chandra et al. provided a complete system that leverage both color and depth information. Their work also provides a new point of view for using depth value in the cutting edge deep learning framework.

In learning based methods, body parts can be classified due to training procedure on large scale of the dataset. However, the noisy or incomplete dataset could cause incorrect classification results. A large amount of training data is required to ensure the robustness of the classification results. For example, in [31], over 1 million frames with different human subjects were used for training purpose. The boundary of different body parts sometimes can be ambiguous, which can cause unstable or incorrect skeleton extraction results.

2.4. Summary

After reviewing the all three different strategies, it is clear that human pose estimation is getting more attention from researchers, and more innovating methodologies are adopted for human pose estimation purpose. However, there are still many open challenges remaining to be addressed. Although some researchers have proposed methods to estimate the 3D locations of joints with machine learning or convolutional neural networks, accurate joint detection is still a new and open challenge. To overcome this issue, we proposed two methods to detect joints in the human body with different strategies. The first method is a hybrid framework that combines human body model and geodesic features of human body together to detect and estimate the position of joints. The second method is a joint detection method

based on data-driven body part segmentation. The proposed body part segmentation method automatically segments the human body into different parts based on the geodesic features of the human body without using a predefined model. Both methods use skeleton model, which is defined as a set of rules, to label the detected body parts and joints.

CHAPTER 3

METHOD

3.1. Overview

3.1.1. Hybrid Joint Detection (HJD)

The proposed hybrid joint detection method categorizes joints into two types, implicit joints, and dominant joints. Implicit joints are defined as the ones close to the torso. In the proposed method, neck, left and right shoulders, left and right hips and waist are defined as implicit joints. Dominant joints are defined as the joints on the limbs of a human body. Left and right elbows and knees are categorized as dominant joints. In practice, the dominant joints are easier to detect than the implicit joins due to the rigidity of the bone segments of the limbs of the human body. Implicit joints are more difficult to detect. Because implicit joints are part of the torso, and the deformation of these joints are less significant than that of the dominant joints. On the other hand, dominant joints carry more information about human motion than the implicit joints. As the connections between torso and limbs of the human body, implicit joints provide information of the overall structure of the human body, such as the width of the torsor and length of the spine. Therefore, it is still necessary to locate the position of the implicit joints.

The hybrid joint detection method employs a skeleton model of human body. The skeleton model defines the geodesic features of implicit joints and the rules to label all the detected feature points (extreme points and joints). Extreme points (feature points on the tip of body parts) are detected and then mapped to the corresponding parts of the skeleton model. Implicit joints are directly located by the skeleton model. The global shortest paths are generated from the centroid of the head to other extreme points are used to provide candidates for domininant joints. Finally, a data-driven method is applied to each limb, to detect possible dominant joints. The dominant joints are determined by voting the results of possible joints from each limb and joint candidates on the global shortest paths. Overall the proposed hybrid joint detection method combines data-driven and model-driven method

together to determine the position of all joints.

3.1.2. Segmentation based Joint Detection (SJD)

Segmentation based joint detection method is inspired by the global shortest path generated from the human body. The features of the shortest path are a good fit for describing the geodesic features as well as the changing between different limbs on the human body. In this method, head detection is also applied first to locate the position of the head. Then extreme points are detected using the same method with HJD. Then for each extreme point, shortest paths between it and the rest of extreme points are generated. In practice, the extreme point of the head is not included, because the region of the head is already detected by the head detection method. For each group of shortest paths that start from the same extreme point are used to compare their motion vectors. When the direction of the vectors is larger than a threshold, it is considered that this group of shortest paths are split, and these paths start to go into different body parts. Then, the segments of different limbs can be obtained by setting the splitting points as the boundaries between limbs and the torso. When limb segments are isolated, the joint detection method is applied on each of them. The advantage of such method is that the searching space for each joint is limited to the corresponding limb. The robustness and accuracy of the system could be potentially improved.

3.2. Background Subtraction

There are many background removal methods [14] proposed to remove the background from depth frames. In both proposed method, the camera remains a fixed position all the time. Therefore, the background is removed by using frame differencing strategy which is widely used for background removal in RGB frames. In the proposed work, instead of calculating the difference of color information between frames, the difference of the distance value is calculated between frames to remove the background. The value of each pixel in depth frame represents its distance to the camera. If the object remains stationary, ideally its depth value remains the same. In reality, however, due to the presence of noise, the

depth value of a pixel varies in a certain range even though the scene remains unchanged. Therefore, the average depth value at a pixel across a temporal range is used to create a smoothed background model. The background model is formed as follows:

$$\forall p_{i,j} \in \mathcal{B},\ p_{i,j} = \sum_{n=1}^{N} p_{i,j}^{n}/N, \tag{3.1}$$

where $p_{i,j}$ is the depth value at a pixel location (i, j) of background model $\mathcal{B}$, $p_{i,j}^{n}$ is the depth value at a pixel location (i, j) of the nth frame, and N is the number of frames that are used to build the background model. For each pixel in the background model, calculate average value of the sum of depth value of all pixels at same position from the sample frames as its depth value. When an object appears in the view, the depth value of pixels in the body of that object changes, and the value changing should be larger than the threshold t. Hence, the foreground can be obtained as following:

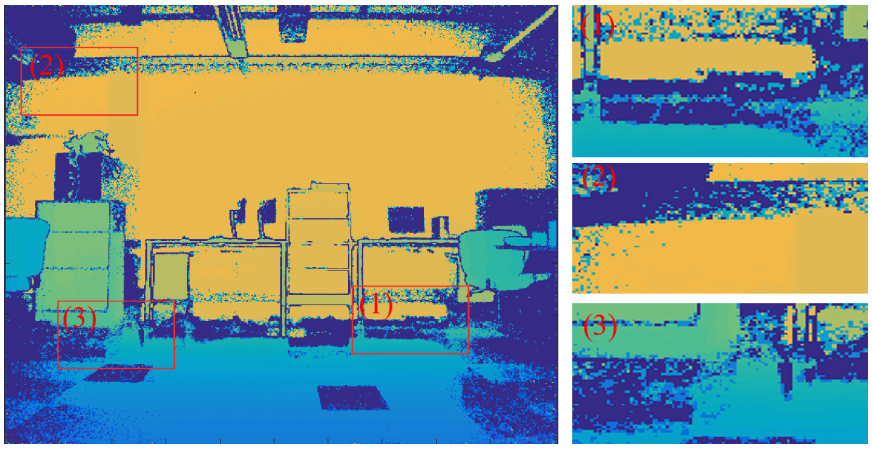
$$\begin{aligned} P^{m} &= \{p_{i,j}^{m} \in f^{m} |\ \triangle\ p_{i,j}^{m} > t\}, \\ \forall p_{i,j}^{m} \in f^{m} \text{ and } &\quad \triangle\ p_{i,j}^{m} = p_{i,j}^{m} - p_{i,j}, \end{aligned} \tag{3.2}$$

where P^{m} is the set of foreground pixels and $\triangle\ p_{i,j}^{m}$ is the difference between pixel $p_{i,j}^{m}$ in frame f^{m} and pixel $p_{i,j}$ in background model. Fig. 3.1 shows the examples of averaged background model with different number of frames. Fig.(a) shows a single frame containing the background. A lot of noise is visible. Fig.(b) shows averaged background model using 20 frames. The amount of noisy pixels are reduced, and the background is more smooth. Fig.(c) shows averaged background model using 70 frames. Number of noisy pixels are significantly reduced, and the background model is smoothed.

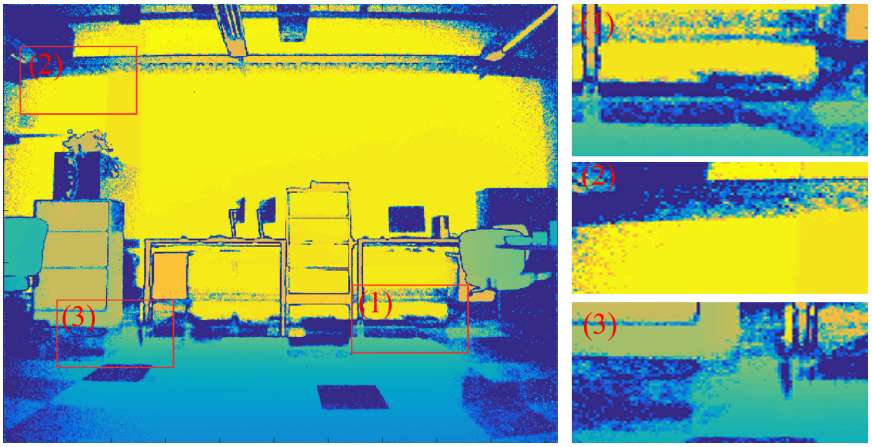
3.3. Extreme Point Detection and Mapping

3.3.1. Extreme Point Detection

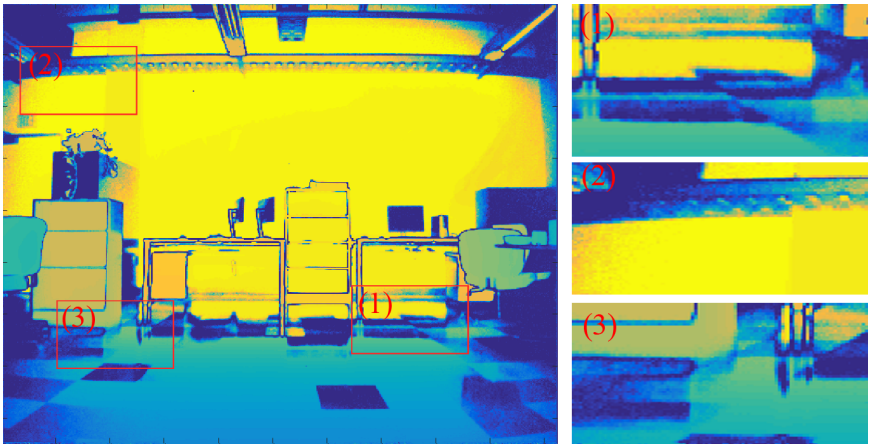
Extreme points on a human body include the head, hands, and feet. As part of the feature points, the spatial distribution of extreme points provides general information of human pose. To detect extreme points, a graph-based method is proposed where the human body is considered as a graph and each point in the point cloud is treated as a node



(a)



(b)



(c)

FIGURE 3.1. Examples of the background model. For each example, three areas are captured and enlarged to show the level of the noise.

on the graph. Each node is connected with its neighbors, if they are on the same surface. Let P denote the 3D point cloud of a human body. The 3D point cloud is calculated from the captured depth frame. Starting from a randomly selected point in the 3D point cloud, denoted as p_0, the geodesic distance between any other point to p_0 is defined as the shortest geodesic distance to p_0. The geodesic distance between a given point to p_0 is calculated as follow:

$$D_g(p_0, P(x,y)) = \sum D_g(P(x_p, y_p), P(x_q, y_q)). \tag{3.3}$$

In the above equation, $P(x_p, y_p)$ and $P(x_q, y_q)$ are neighboring points on the shortest path between $P(x, y)$ and p_0. $D_g(\cdot)$ represents the geodesic distance between two points on the point cloud P. To calculate the distances from each point on the point cloud to p_0, we adopted an iterative way to go through the whole point cloud. Start from p_0, each point on the point cloud calculate the distance between itself to its eight nearest neighbors. The shortest distance from each of the eight neighbors to the p_0 will be updated. For a new point, its distance value is calculated; for a point, whose distance to p_0 already has been calculated, the distance is updated when a shorter distance is found. A distance map is generated when all points on the point cloud have been calculated, and a point with the longest distance (extreme point), denoted as E_1 is represented as follows:

$$E_1 = arg\max D_g(p_0, P(x,y)). \tag{3.4}$$

Algorithm. (1) explains the steps to generate a distance map and find an extreme point. To avoid the same extreme points being repeatedly found, when an extreme point is identified, its geodesic distance to any existing extreme point is set to zero. Therefore, when a new extreme point is found, it must have the longest geodesic distance to all the existing points. Thus, five distance maps are usually required. Let M^i denotes the distance map, where i is the index of distance map. The final updated distance map is as follows:

$$M(x,y) = min(M^1(x,y), M^2(x,y)...M^n(x,y)). \tag{3.5}$$

Algorithm 1 Iteratively update distance map

Randomly select a point p_0 on the point cloud P.

Push p_0 into queue Q_1.

while Q_1 is not empty **do**

  for each point p_i in Q_1 **do**

    for each neighbor p_j^i of point p_i **do**

      if p_j^i and p_i are on the same surface AND p_j^i has not been updated **then**

        $D_g(p_j^i, p_0) = D_g(p_i, p_j^i) + D_g(p_i, p_0)$.

      else if p_j^i and p_i are on the same surface AND p_j^i has already been updated **then**

        $D_g(p_j^i, p_0) = min[D_g(p_i, p_j^i) + D_g(p_i, p_0), D_g(p_j^i, p_0)]$.

      end if

      Push p_j^i into queue Q_2.

    end for

    Remove p_i from Q_1.

    if Q_1 is empty **then**

      $Q_1 = Q_2$.

      Clear Q_2.

    end if

  end for

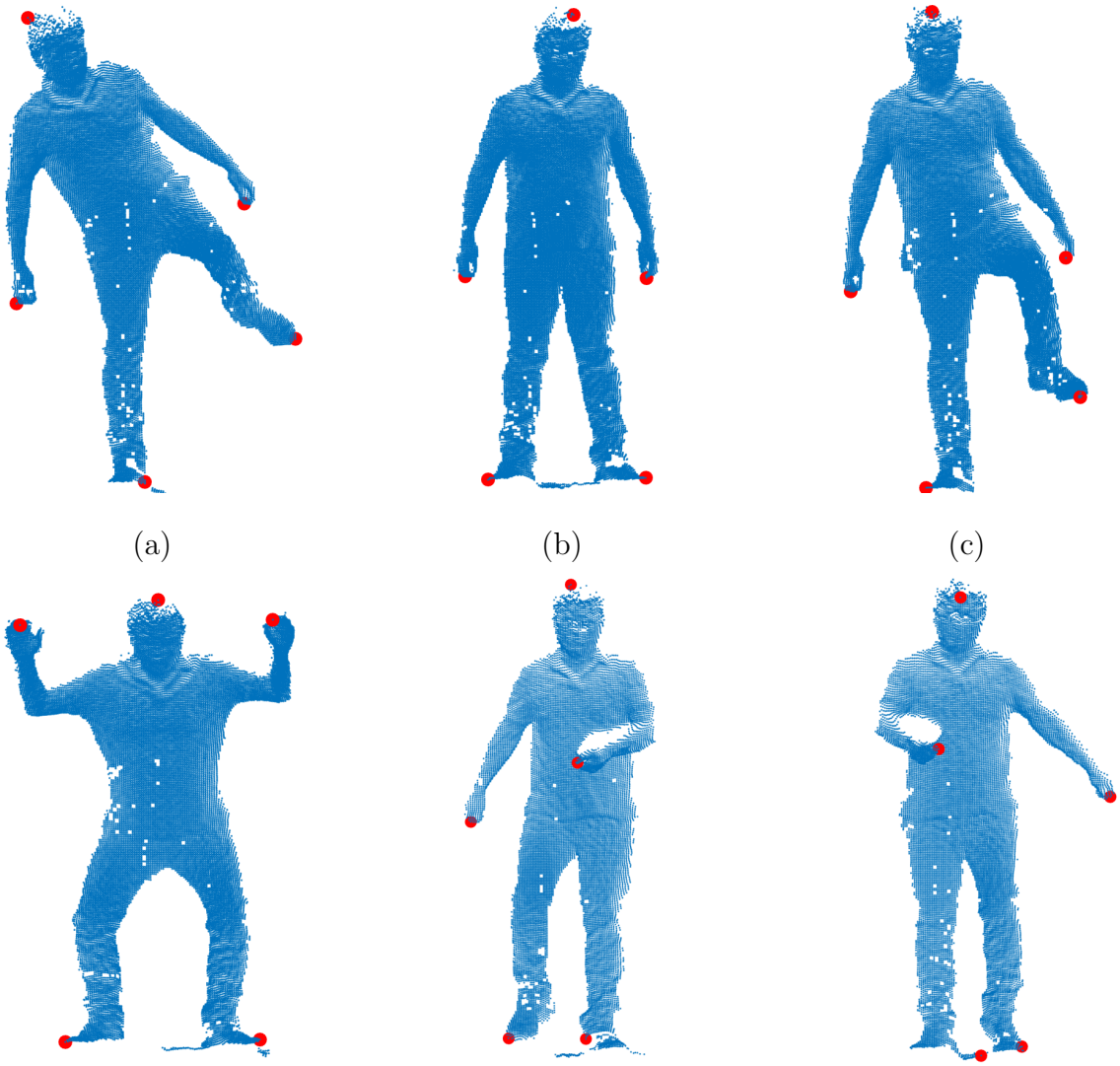
end while

Select the point with the largest distance as an extreme point.

Furthermore, Eq. (3.4) is rewritten in a more general form:

$$E_i = arg\max D_g(E_{i-1}, P(x, y)), i > 0. \tag{3.6}$$

However, an issue was discovered when we are conducting the experiments. When detecting extreme points, the head cannot be detected all the time due to the deformation of the human body. To overcome this problem, head detection is applied before the detection of extreme points. Then, the detection of extreme points starts from the centroid of the



(a) (b) (c)

(d) (e) (f)

FIGURE 3.2. Examples of extreme points in different poses. The red dots indicate the position of the extreme points. From (a) to(d) extreme points in poses without self occlusion are presented. Figure(e) and (f) show the extreme points in poses with self occlusion. In both (e) and (f) the large holes in the torso are caused by the arms in front of the torso. In (f), extreme point on the foot on left side is misplaced due to the noise caused by reflection on the floor.

head instead of a random point.

Self-occlusion usually happens in different motions and activities. For example, when a person drinks water, his or her arm will block part of the torso. To handle the self-occlusion, we compute the difference of the depth value between adjacent points when a distance map is updating. If the difference is less than a threshold δ, the two points are considered as lying on the same surface of the human body; otherwise, they are considered as belonging to different body parts. If the two points belong to different body parts, the edge between them will be disconnected. Thus, adjacent points from a different surface cannot update their distance to the initial point from each other. Therefore, at any point in P, its geodesic distance is only updated according to its neighboring points on the same body part. Fig. 3.2 shows two examples of the results of extreme point detection.

3.3.2. Extreme Point Mapping

When extreme points are detected, there is no correspondence between extreme points and body parts on the skeleton model. Without knowing the correspondence between extreme points and the skeleton model, it is difficult to detect the positions of joints. Thus, mapping the extreme points to the human body model ensures that the data-driven method works with the human body model. The mapping method starts from mapping an extreme point to the head. The head is detected by a trained classifier using linear discriminant analysis (LDA). To map the other extreme points to the human body model, the geodesic relationship between hands and feet is used. In the skeleton model, the geodesic distance between the head and the hands are shorter than the geodesic distance between the head and the feet, that is

$$D_g(p_{head}, p_{hand}) < D_g(p_{head}, p_{foot}). \tag{3.7}$$

With the above constraints, the extreme points of feet and hands are separated. To determine if an extreme point of hand corresponds to the left or right hand, we assume that the geodesic distance between the left hand and the left shoulder is shorter than the geodesic distance between the left hand and the right shoulder, and the same logic is applied to the right hand.

The relationship between the left and the right hands can be described as follows:

$$D_g(p_{Lh}, j_{Ls}) < D_g(p_{Lh}, j_{Rs}),$$
$$D_g(p_{Rh}, j_{Rs}) < D_g(p_{Rh}, j_{Ls}), \tag{3.8}$$

where p_{Lh} and p_{Rh} represent the extreme points of left and right hands, respectively, p_{Ls} and p_{Rs} represent the estimated joints of left and right shoulders, respectively. Estimating the position of shoulders is presented in Section 3.5.2. The relationship between the left and the right hands is also suitable for the left and the right feet.

3.4. Head Detection

The objective of head detection is to initialize the extreme point mapping process. The position of the head is used as the initial landmark during the extreme point mapping. After the head is mapped, the other extreme points are mapped according to the rules defined in the skeleton model. To detect the head, the method proposed in [10] is adopted and modified in this work. In our proposed work, feature vectors containing depth features around the center area of the head is used to train and detect the head. To obtain the feature vector of the head, a point p_c in the center area of the head is selected first. Then, select a set of points P_c on the circle around the p_c are selected, and the distance difference between each point in P_c and the point p_c is calculated. The calculated distance difference is sorted and stored in the feature vector. In [10], two set of points around the p_c are selected to form the feature vector. One set of points are selected within the region of the head, another set of points are selected outside the region of the head. Because the background is not removed from the depth frame in [10], therefore the points outside the region of head provide extra depth information between head and background. In our work, only the points within the region of the head are used due to the removal of background.

To detect the head region, a trained classifier was applied to every pixel in depth frames to label the pixels. A false positive filter was developed to eliminate most of the points that do not belong to the head. In our method, the trained classifier is applied only in the areas around each extracted extreme point. Geodesic distance is used to select the points

to form the areas around each extreme point so that only the points are geodesically close to the extreme points are selected. As a result, the time for head detection is significantly reduced. After the all the areas are labeled, the area with most positive labels is considered as the region of the head, and the corresponding extreme point is labeled as the head extreme point. Then, the centroid of the head is calculated by the averaged position of all the positive pixels in the region of the head. Figure. (3.3) shows the exemplary classification results for head detection.

The position of the head is also tracked over time. In case that extreme points are missing or incorrectly detected, tracking the position of the head allows the system still correctly locate the position of the head.

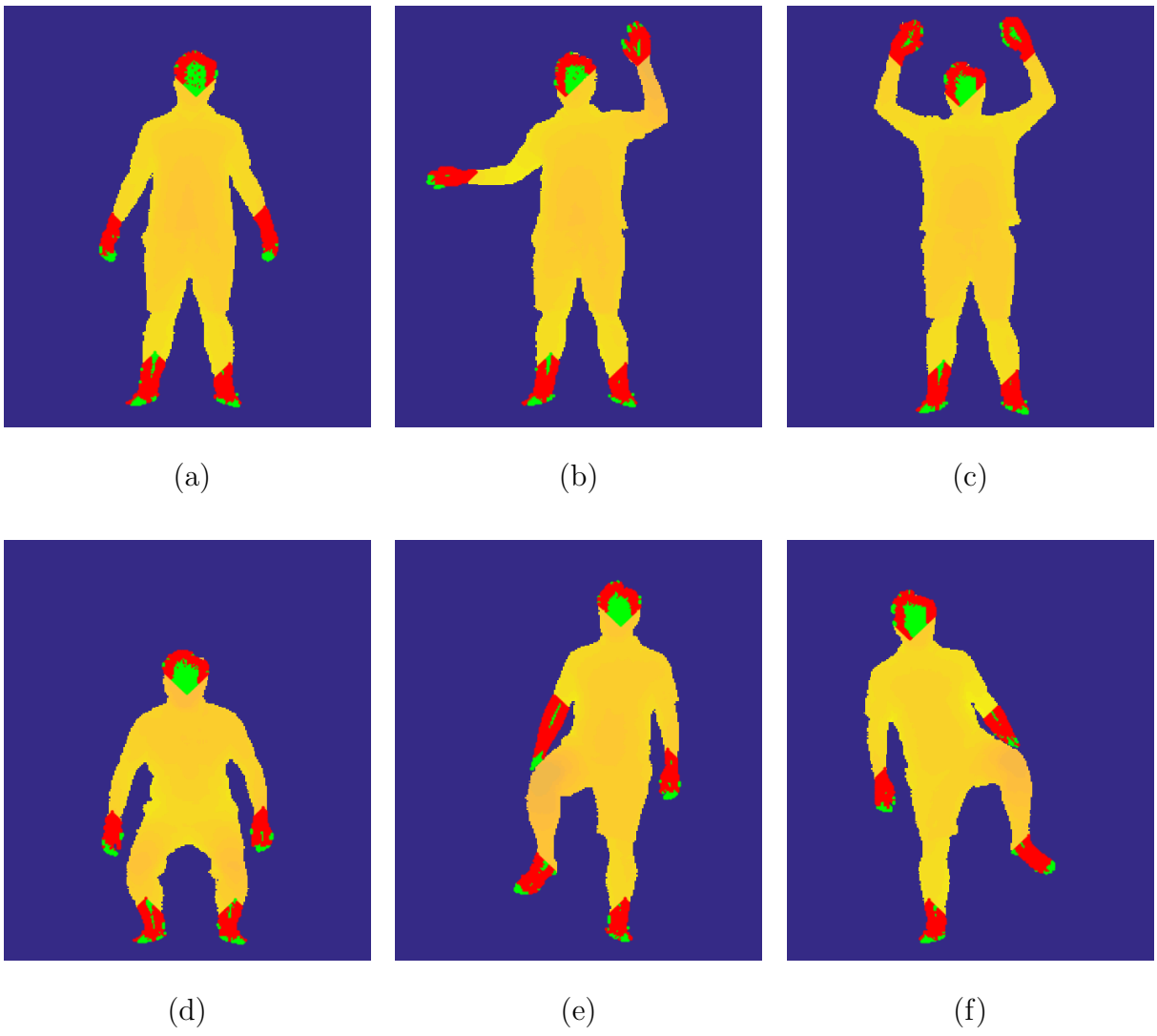
3.5. Hybrid Joint Detection (HJD)

3.5.1. Skeleton Model

As part of the proposed hybrid framework, the skeleton model estimates the positions of the implicit joints and provide constraints for data-driven joint detection algorithm. The traditional model-based human pose detection methods [45, 46, 40, 20] define the human body model with a collection of body parts and DOFs (degrees of freedom) or joints with articulated structure and DOFs of joints. Our method defines the human body model only by defining the overall structure and general geodesic features of the human body model. Such skeleton model provides more space to optimize the locations of implicit joints and more flexibility to fit into different body shapes. For implicit joints, relative position and size are defined. On the other hand, the only relative position is defined for each dominant joint. Fig. 3.4 shows the skeleton model used in our method. In this figure, there are three types of point, the green points represent the extreme points, the blue points represent the implicit joint, and the red ones represent the dominant joints.

3.5.2. Estimation of the Implicit Joints

The implicit joints as part of the torso are more difficult to detect than the dominant joints such as elbows. However, because implicit joints locate within the torso, they



(a) (b) (c)

(d) (e) (f)

FIGURE 3.3. Examples of classification results for head detection with different poses. The green pixels are classified as head. The red pixels are not classified as head. The region with the most positive pixels is considered as the head.

have much-limited DOFs than the dominant joints. As a result, the model-based estimation methods provide reliable results for estimating the position of implicit joints. When estimating the position of implicit joints, we take full advantage of the geodesic features of the human skeleton to focus on the possible positions of joints. The estimation procedure follows a top-to-bottom order. The position of the neck is estimated first based on the position of

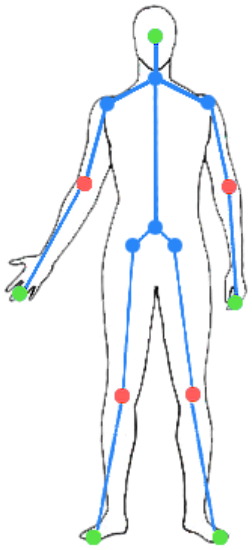


FIGURE 3.4. The skeleton model used in our method. The green dots represent the extreme points. Blue dots represent implicit joints (neck, waist, shoulders and hips). Red dots represent dominant joints (elbows and knees).

the head. Given the length between neck and shoulder, denoted as l_{ns}, the left and right shoulders are defined as follows:

$$\{p_i \mid p_i \in P; D_g(p_{neck}, p_i) = l_{ns}; D_g(p_{head}, p_i) > D_g(p_{head}, p_{neck})\} \tag{3.9}$$

$$\{p_j \mid p_j \in P; D_g(p_{neck}, p_j) = l_{ns}; D_g(p_{head}, p_j) > D_g(p_{head}, p_{neck})\} \tag{3.10}$$

$$and, i \neq j$$

$$p_i, p_j = arg\max_{i,j}(A(p_i, p_{neck}, p_j)). \tag{3.11}$$

In the above definition, p_i and p_j are two points in P, $A(\cdot)$ is the function to calculate the Euler angle between p_i and p_j. Eq. (3.11) ensures left and right shoulder are separated as much as possible. The hips are defined in a similar way to the shoulders because the structure of neck-shoulders and waist-hips are both triangle structure based on the skeleton structure of the human body. Thus, given the distance between the waist and the hips l_{wh}, the hips are defined as follows:

$$\{p_m \mid p_m \in P; D_g(p_{waist}, p_m) = l_{wh}; D_g(p_{head}, p_m) > D_g(p_{head}, p_{waist})\} \tag{3.12}$$

$$\{p_n \mid p_n \in P; D_g(p_{waist}, p_n) = l_{wh}; D_g(p_{head}, p_n) > D_g(p_{head}, p_{waist})\} \tag{3.13}$$

$$and, m \neq n$$

$$p_m, p_n = arg\max_{m,n}(A(p_m, p_{waist}, p_n)). \tag{3.14}$$

Here, we assume that the geodesic distance from head to any shoulder is greater than that of the head to the neck, and the geodesic distance from head to any hip is greater than that of the head to the waist. The waist is defined as:

$$p_{waist} \in \{p_k \mid D_g(p_{head}, p_k) = l_w; |D_g(p_{Ls}, p_k) - D_g(p_{Rs}, p_k)| < \mu\}, \tag{3.15}$$

where l_w is the given distance from the head to the waist, μ denotes the threshold of the difference between the geodesic distance from the left and the right shoulder to the waist. The skeleton model requires the waist to have a close distance to the left and right shoulders. This ensures the scope of the waist locates within the torso instead of arms. Fig. (3.5) illustrates the process and constraints for estimating the positions of implicit joints.

3.5.3. Detection of the Dominant Joints

On the human body, elbows and knees are defined as dominant joints. In the proposed system, a data-driven method is used to detect these joints because dominant joints usually cause more significant deformation of the limbs in contrast to the implicit joints. In this work, a method that integrates two detection strategies is developed to ensure accurate and stable detection results. A global shortest path based strategy is employed to detect candidates for the dominant joints, and a local detection for each elbow and knee is employed. The detection results of elbows and knees are averaged results from both the shortest path based method and specific detection method.

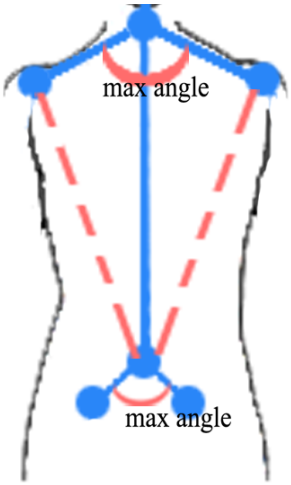


FIGURE 3.5. Illustration of the constraints for estimating shoulders, waist, and hips.

The global shortest path based method uses the distance map similar to the distance maps used in Section 3.3.1. The distance map starts from the centroid point of the head, denoted as p'_{head}, and calculate the geodesic distance to all the other points in the point cloud. During the procedure of updating the distance map, each updated point is connected to the neighbor that has the closest geodesic distance to the starting point (centroid of the head). Therefore, the shortest paths from p'_{head} to all extreme points can be generated during the updating procedure of the distance map. For each shortest path, given the start and end points of a path p_{start} and p_{end}, the joint candidates on it should satisfy the following condition:

$$p_i, p_j ... p_n = \arg\min_{i,j...n}(D_g - D_e), \tag{3.16}$$

$$D_g = D_g(p_{start}, p_i) + D_g(p_{start}, p_j) + ... + D_g(p_n, p_{end}),$$

$$D_e = D_e(p_{start}, p_i) + D_e(p_{start}, p_j) + ... + D_e(p_n, p_{end}).$$

The objective is to minimize the difference between the cumulative Euclidean distance and the geodesic distance of the path. To limit the number of joint candidates, the following

restrictions are enforced:

$$\forall p_i, A(p_i) < \beta, R_g(p_i) > \alpha, \tag{3.17}$$

where $A(p_i)$ represents the Euler angle formed by p_i and its two adjacent points p_{i-1} and p_{i+1}, and β and α are defined as threshold variables. The $R_g(p_i)$ is the geodesic distance ratio on p_i, defined as:

$$R_g(p_i) = \frac{\min(D_g(p_{i-1}, p_i), D_g(p_i, p_{i+1}))}{D_g(p_{i-1}, p_i) + D_g(p_i, p_{i+1}))}. \tag{3.18}$$

The restrictions ensure that the joint candidates show how curvy the path is, and the points that close to the endpoints of the path are not found as candidates. Because the sharper the angle is and the greater the geodesic distance ratio is, the more contribution of the corresponding joint candidate makes to bend the limb. Algorithm. 2 shows the detail of selecting joint candidates from global shortest paths. An example of the shortest path from the head to the other extreme points is shown in Fig. (3.6).

The objective of the local joint detection is to detect the most possible joint positions for each limb. A local shortest path from the corresponding extreme point to its closest implicit joint (e.g. shoulder or hip) is created. For example, the shortest path from the left hand to the left shoulder is created for detecting the position of the left elbow. Given the start and end points p'_{start} and p'_{end} of the shortest path on each limb, the detected joint must satisfy the following conditions:

$$p_k = \min_k (D_e(p'_{start}, p_k) + D_e(p_k, p'_{end})). \tag{3.19}$$

$$A(p_k) < \beta, A(p_k) = \angle p_{start} p_k p_{end}. \tag{3.20}$$

This is to prevent random detection when the limb stretches straight. The position of each dominant joint on limbs is the average position of the joint candidates on the corresponding limb from Eq. (3.16) and the detected joint from Eq. (3.19). Furthermore, when

Algorithm 2 Selecting joint candidates on a shortest path

For a given path L, its start and end points are denoted as p_start and p_end. p_i represents a point on L between p_start and p_end.

Q_p is used to store selected joint candidates, Q_L is the queues to store sub-paths of L.

Push L into Q_L.

while number of joint candidates $<$ threshold **do**

  for each path L_i in Q_L **do**

    Find p_i, so that $p_i = \arg\max_i(D_e(p^i_{start}, p_i) + D_e(p_i, p^i_{end}))$.

    Push p_i into Q_p.

    Remove L_i from Q_L.

    p_i divides L_i into two sub-paths L_j and L_k.

    Push L_j, L_k into Q_L.

  end for

end while

for each p_j in Q_p **do**

  if $\angle p_i < \beta$ **then**

    Remove p_i from Q_p.

  end if

  if $R_g(p_i) < threshold$ AND $R_g(p_{i+1}) < threshold$ **then**

    Remove the point with larger angle.

    Re-caculate the Geodesic Ratio for the remaining point.

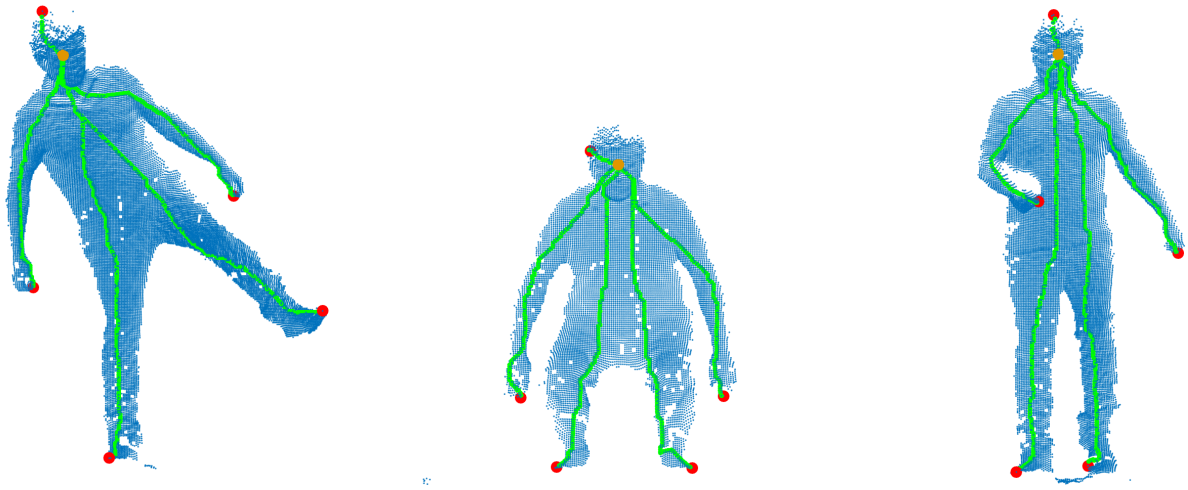
  else if $R_g(p_i) < threshold$ AND $R_g(p_{i+1}) > threshold$ **then**

    Remove p_i from Q_p.

  end if

end for

Return Q_p as the set of joint candidates on the given path L.



(a) (b) (c)

FIGURE 3.6. Examples of the shortest paths. All shortest paths are start from the centroid of head, which is the yellow dot. Extreme points are represented by red dots. The green dots represent the shortest path.

a dominant joint cannot be detected, a geodesic middle point on the shortest path of the corresponding limb is used instead. Examples of the shortest path for the specific detection are shown in Fig. (3.6).

3.6. Segmentation based Joint Detection(SJD)

A body segmentation based joint detection method is also proposed in this work. The goal of body part segmentation is to divide the human body into the torso, arms, and legs. The highlight of this method is that a local data-driven strategy is applied to detect the boundaries between limbs and the torso without using any model or learning procedure. To detect the boundaries between limbs and the torso, the shortest paths on the human body are still used. For each limb, shortest paths start from the corresponding extreme point to the other extreme points are selected. It is assumed that there is not much variety in directions across all the shortest paths for the parts inside the limb. For the parts outside of the limb, the directions across all shortest paths become more varied because all the shortest path are ended to different extreme points which belong to different body parts.

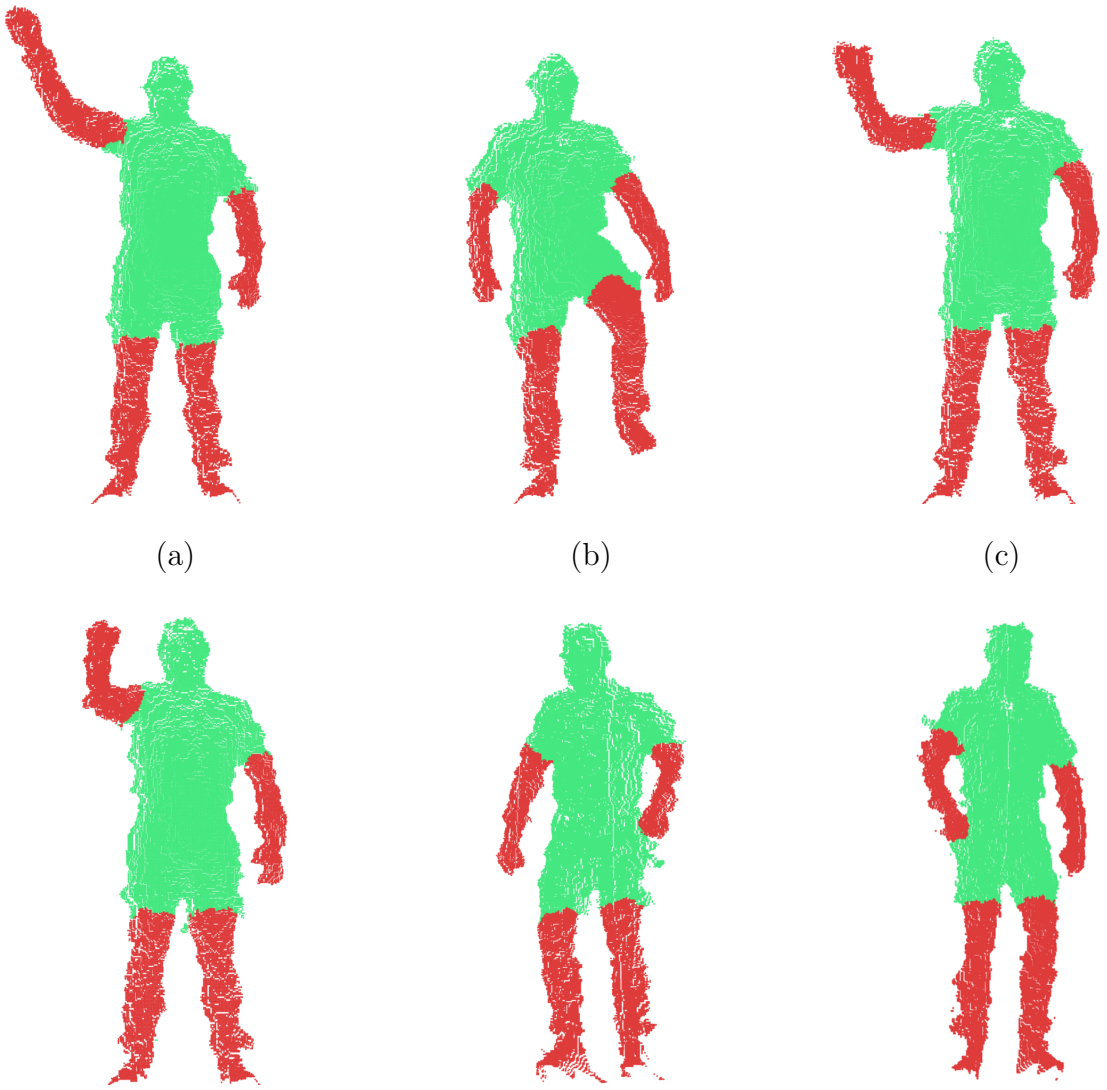
3.6.1. Human Body Segmentation

For a given extreme point $E_i (E_i \in E)$, a updated distance map M_i is generated starting from E_i. Shortest paths from E_i to the rest of extreme points E_j in E are generated by using distance map M_i. On each shortest path L_{ij}, a set of vectors v_m^{ij} $(m = 1, 2...M)$ are calculated to indicate the direction of each corresponding part on the shortest path. Then calculate Euler angles between every two vectors across all the shortest path. When the Euler angle between a pair of vectors becomes larger than the threshold β^*, the vectors are selected as the breaking vectors. The endpoint with less distance value of these two vectors is selected as the breaking point. The points on the human body with same geodesic distance to the extreme point E_i are selected as the boundary between the limb and torso. Algorithm 3 shows the steps of the procedure of segmentation. Exemplar segmentation results are shown in Fig. 3.7.

3.6.2. Dominant Joint Detection

After the human body is segmented, dominant joint detection is applied in each limb. A likelihood function is applied to each point on the shortest path within the corresponding limb. Unlike the method in HJD, only local information and features of each limb are used to detect the dominant joints. The global information is not involved in the dominant joint detection stage. Two features are used to determine the likelihood of each point. The first feature is the Euler angle formed by the selected point and two endpoints on the shortest path. The second feature is the distance between a selected point and the ideal position of the joint. The ideal position of a joint on a limb is determined by calculating the average position of each joint across the dataset. The Euler angle indicates that how much a point contributes to bending the shortest path. The second feature indicates how well a point is placed compare with the ideal position. The likelihood function for detecting dominant joints is defined as follow:

$$L(p_i) = cos(A(p_i)/2) + R(p_i), \tag{3.21}$$



(a) (b) (c)

(d) (e) (f)

FIGURE 3.7. Examples of body segmentation with different poses from IDT 13 dataset [16]. The limbs are colored with red, and the torso is colored with green.

$$R(p_i) = \frac{1}{e}\frac{D_g(p_i, p_{end})}{D_g(p_{start}, p_{end})}^{-r}. \tag{3.22}$$

In equation 3.21, $L_(p_i)$ represents the likelihood of being a joint for p_i on its shortest path. $A(\circ)$ represents the Euler angle formed by p_i and the two endpoints on the shortest

path. $R(p_i)$ describe how close the selected point p_i to the ideal joint location. In equation 3.22, r is a constant value indicates the geodesic distance ratio. The geodesic distance ratio is calculated by letting the geodesic distance of a shortest path divided by the geodesic distance between the endpoint (an extreme point is also an endpoint) and the joint on the same shortest path.

Algorithm 3 Segmentation on human body

Here, len represents the size of vectors.

for each extreme point E_i, exclude the extreme point of head **do**

for each shortest path L_{ij} from E_i to E_j AND $i \neq j$ **do**

for point p_n on L_{ij} **do**

Calculate the vector v_{ij} between $p_n and p_{n+len}$.

Push v_{ij} into array Q_{ij}.

end for

end for

for each vector v_m^{ij} in each Q_{ij} **do**

Calculate Euler angle between every two vectors with same index m.

if Calculated Euler angle ¿ β^* **then**

p_{len*m} on all the shortest path L_{ij} are pushed into Q_p.

for each point in Q_p **do**

Select the point with shortest geodesic distance to E_i, denoted as p^i.

for each point q_j on human body **do**

if $D_g(q_j, E_i) < D_g(p^i, E_i)$ **then**

q_j is labeled as a point in the limb corresponding to extreme point E_i.

end if

end for

end for

end if

end for

end for

CHAPTER 4

EXPERIMENTS AND EVALUATION

4.1. Dataset

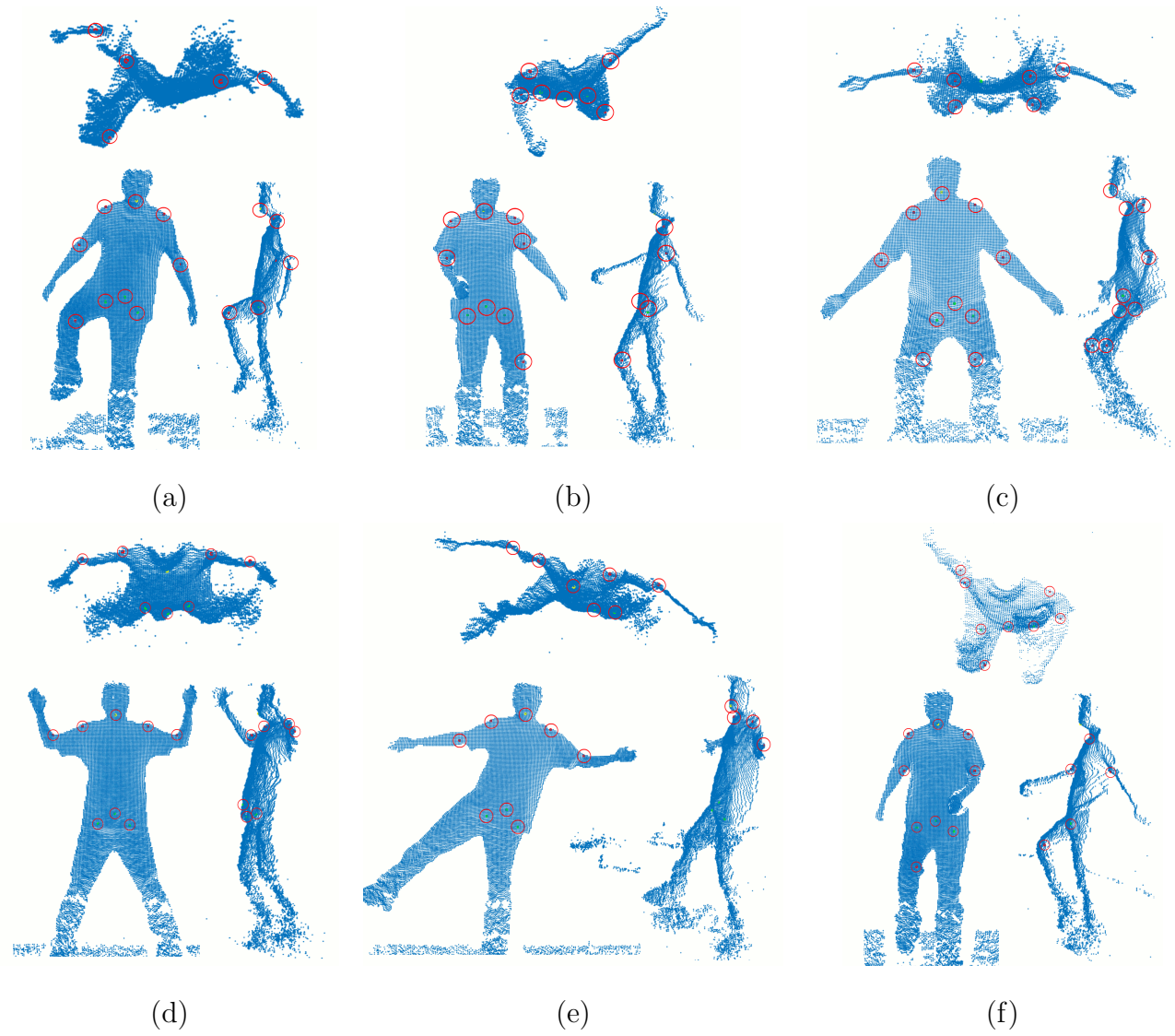
Human pose detection from depth videos is still a relatively new topic. Therefore, the amount of available public datasets is very limited. According to the latest survey [43], [6], there are datasets were collected without the ground truth of feature points. There are also datasets with limited poses. Such datasets are collected mainly for action recognition or detection for certain activities. For the purpose of estimating the poses of a human subject, ideal dataset should provide the ground truth of all feature points and poses with different human subjects. Among all available public datasets, few datasets provide ground truth for all feature points. To have ground truth labeled, there are two ways: labeling the ground truth manually and labeling the ground truth by using marker based motion capture system. Manually labeling all the feature points is the most straightforward way to label the ground truth. It is easy to be implemented and to be adopted by most of the researchers, due to its low cost. On the other hand, the latter requires not only a complex hardware system but also a specialized room to be fully functional.

In this dissertation, we first recorded our own dataset with the latest Microsoft Kinect camera, and manually labeled all the joints. Then, we adopted the IDT 13 dataset from [16]. The IDT 13 dataset contains 6 sequences with several different poses by the same person. The ground truth of each feature point (extreme point and joint) is captured by a commercial motion capture system. The device used for IDT 13 is Swiss Ranger 4000 depth camera, which has 176×144 pixels. Comparing with Swiss Ranger 4000, Microsoft Kinect has a much higher resolution (512×424 pixels) with better accuracy.

4.2. Evaluation of Hybrid Joint Detection (HJD)

To evaluate the proposed hybrid joint detection method, 10 videos were recorded using Microsoft Kinect camera. The acquired videos contain various human poses such as walking, kicking, turning the upper body, and jumping. The reference points for joints

are manually annotated in the 3D point cloud. Examples of the detection are depicted in Fig. 4.1, and three different views are shown for each result to give a 3D view of the joints.



(a) (b) (c)

(d) (e) (f)

FIGURE 4.1. Examples of detection results. The detected joints are marked with red circles. Each result is displayed in three views: top view at the top, front view at the bottom left and side view at the bottom right.

4.2.1. Detection Rate

Because the proposed method focuses on joint detection, it is necessary to evaluate the detection rate of joints. In this evaluation, if a point on the human body is detected and

mapped to a joint, then the joint is considered to be detected without considering the error distance to the ground truth. Table 4.1 lists the overall detection rate of the implicit joints.

TABLE 4.1. Detection Rate of Implicit Joints (%)

Feature Points		Shoulder		Hip	
	Neck	left	right	left	right
Detection Rate	88.3	88.0	88.5	83.6	83.1

Since the implicit joints are mostly estimated by the human skeleton model, the failure cases are mostly caused by the inaccurate head detection. The detection rate of hips is slightly lower than that of the shoulders because hands and other body parts occluded the hips in some of the frames in the data set. Geodesic features are used when estimating the implicit joints by the skeleton model, the areas of the hip with the corresponding geodesic distance value is not detectable when the areas are occluded by other body parts.

TABLE 4.2. Detection Rate of Dominant Joints (%)

Feature		Elbow		Knee	
Points	Waist	left	right	left	right
Detection Rate	86.7	90.1	89.3	89	90.2

The detection rate of the dominant joints is greater than that of the implicit joints partly due to the shortest path based and specific detections. In Table 4.2, we discuss the situations of the significant deformation occurring in the joints area. In practice, when the Euler angle of a bent limb is shaper than 145°, it is considered as significant deformation, which can be detected by the proposed method. It is assumed that when a limb is fully stretched straight, the Euler angle on the corresponding dominant joint is 180°.

4.2.2. Accuracy of Joint Detection

In our evaluation, if a joint is within 6 cm of the selected reference point, then the detection is considered correct. The overall accuracy of all joints are listed in Table 4.3.

TABLE 4.3. Overall Accuracy of Joints (%)

Feature Points			Shoulder		Elbow		Hip		Knee	
	Neck	Waist	left	right	left	right	left	right	left	right
Detection Rate	81.3	86.7	88.3	88	87.2	86.3	83.6	84.1	84	86

In Table 4.3, the accuracy of implicit joints(neck, waist, shoulders, and hips) are close to their detection rate. Because in the proposed method, the skeleton model finds the most suitable points for shoulders and hips after the geodesic constraints are calculated. Comparing to fixed structure human body model, our model can reduce the error distance for shoulders and hips. On the other hand, the overall accuracy of dominant joints is lower than their detection rate. Because when an elbow or knee is not detectable, a geodesic middle point is placed, and the middle points have bigger error distances. A phenomenon that we realized from the experiments is that the deformation of the cloth on the testing object could affect the detection of shortest paths. Therefore, the deformation of cloth could affect the accuracy of dominant joint detection. As a result, only major joints are detected, and minor joints such as ankles and wrists are left behind in the proposed method to ensure the accuracy.

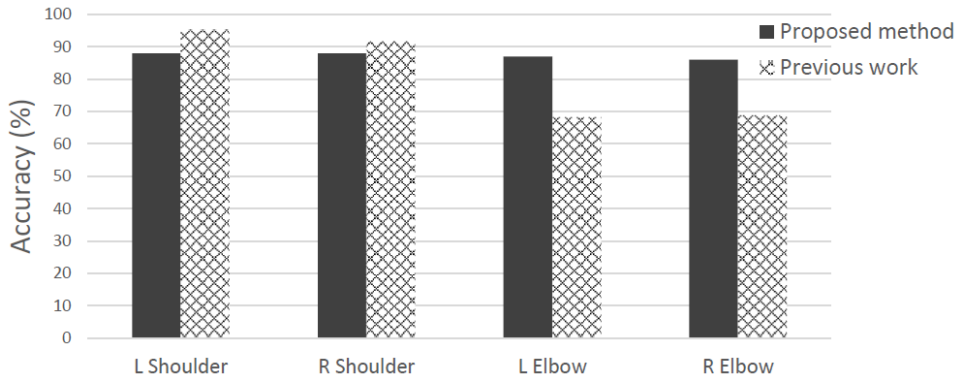


FIGURE 4.2. Comparison between our previous work [42] (shadow bars) and the proposed method(solid bars).

We compare the proposed method against with our previous work [42], and the accu-

racy of elbows in the proposed method is 17.8% higher than our previous work. However, we also realize that the accuracy of shoulders is 5.4% lower. The Fig. 4.2 shows the comparison of the accuracy of elbows and shoulders between [42] and the proposed method. Because the method in [42] only detects shoulders and elbows as the result of joint detection, only the comparison data of elbow and shoulders are listed in Fig. 4.2. The major factor that causes the drop of accuracy on shoulders is that a general skeleton model is used in the proposed method. The accuracy of estimation of shoulders is affected by the detection of the head. In [42] a specific head-shoulder template is used to detect the positions of head and shoulders. Comparing the two different type of models, head-shoulder template can detect head more accurately than the ellipse head model, but it also produces large error distance in some cases, especially when the testing object give complex poses.

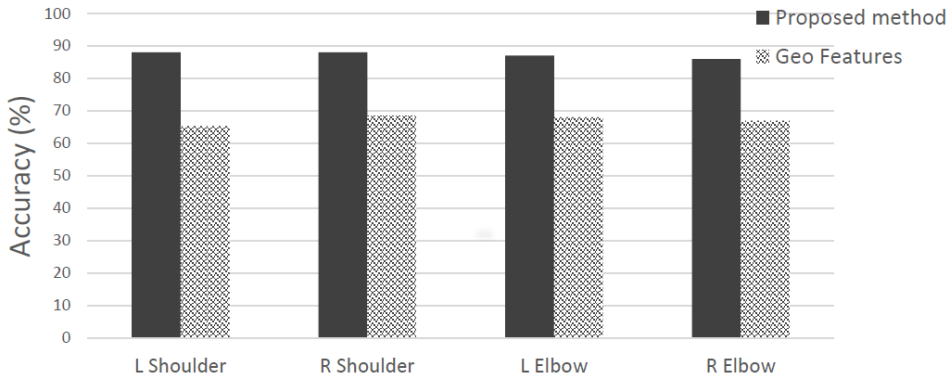


FIGURE 4.3. Comparison between the method in [15] (shadow bars) and the proposed method(solid bars).

We also run the method of [15], which combines model-based estimation and data-driven detection to extract human poses to compare with our method. Because in [15] only shoulders and elbows are detected, only the accuracy of shoulders and elbows are listed in the Fig. (4.3). The average accuracy of elbows and shoulders of our method is 21.79% higher than the accuracy of [15]. In [15], the positions of shoulders are estimated by calculating the average position of selected points with a certain distance to the head and centroid of the torso, fixed searching range is defined for selecting points. In our method, an adaptive

skeleton is applied, therefore the accuracy of shoulders is higher. When detecting elbows, the shortest paths provide more accurate joint candidates to choose, and shortest paths have fewer chance to be affected by the edges of the clothes on human bodies. Comparing to [15], the average accuracy of elbows in our method is 19.25% higher.

4.2.3. Error Distance

Error distance is calculated as the Euclidean distance between detected points and reference points. On a small area on the surface of the human body, it is close to the geodesic distance.

TABLE 4.4. Detection Rate of Implicit Joints (%)

Feature Points	Neck	Shoulders	Elbows	Waist	Hips	Knees
Avg Err Distance(cm)	4.2	4.1	3.3	5.2	5.5	4.2
Max Err Distance(cm)	6	6.1	6.8	7.4	8.8	6.7

The average and the max error distance and listed in Table 4.4. The average error distance of waist and hips are higher than the neck and shoulders, due to the cumulative error caused by the model. Furthermore, hips have no clear boundary on the human body, but shoulders have the clear boundary, which makes them easier to find. Elbows and knees have smaller average error distance than the dominant joints, due to the mixture of two detection methods. The average and max error distances of the shoulders of [15] are 5.7cm and 10.2cm, respectively, and for elbows, the average and max error distances are 4.8cm and 10.1cm, respectively. The max error distance in [15] is mainly caused by the deformation of the edges of clothes.

4.2.4. Analysis of Parameters

In our analysis of parameters, 200 frames that contain 15 different poses were used. In our proposed method, a threshold δ is used to verify if two adjacent points belong to the same surface. Table 4.5 lists the average accuracy of joint detection with different δ values. It is clear that the system achieves the highest accuracy (87%) among all possible thresholds

when δ is at 45 mm. When a lower threshold is used, more points on the same body surface are mistaken as points on the different body surface. On the other hand, as this threshold is increased, points on a different surface are considered to be on the same surface, which, as a consequence, degrade the accuracy. The choice of threshold δ affects the procedure of updating the distance map and, hence, it influences the accuracy of detecting both implicit and dominant joints. In the rest of our experiments, the threshold δ is 45 mm.

TABLE 4.5. Accuracy of detecting joints with different δ(%)

δ(mm)	15	25	35	45	55	65	75	85
Accuracy(%)	15.21	77.17	65.21	87.67	67.93	65.21	58.69	59.29

Another threshold used in our method is θ for selecting candidates for dominant joints, which is the angle of the two vectors formed by three adjacent points. The three adjacent points are selected by the geodesic distance ratio. In general, a small angle allows a fewer number of candidates to be selected. We conducted experiments with different θ and evaluated the average accuracy and detection rate as shown in Fig. (4.4). As θ increases, the detection rate increases, and best detection rate was achieved with θ at 145° and 175°. The accuracy, however, varies fluctuated with the increment of θ. When θ was at 95° and 105°, the accuracy reached nearly 100%. This is due to the low detection rate. Within a few successful detections, the joints were accurate. By considering both detection rate and accuracy, we set θ to 145° in the rest of our experiments.

4.2.5. A Comparison Study with Microsoft Kinect SDK

We conducted a comparison study with Microsoft Kinect SDK following the study in [36] and evaluated the accuracy and consistency of our proposed method. Fig. 4.5 illustrates the average error distance of the detected joints. The error distance is measured with respect to the ground truth marked manually on the acquired data. It is shown that the average error distance of the detection of joints using Microsoft Kinect SDK is 11.56cm; whereas that of our proposed method is 3.36cm. The largest errors in the results of SDK are

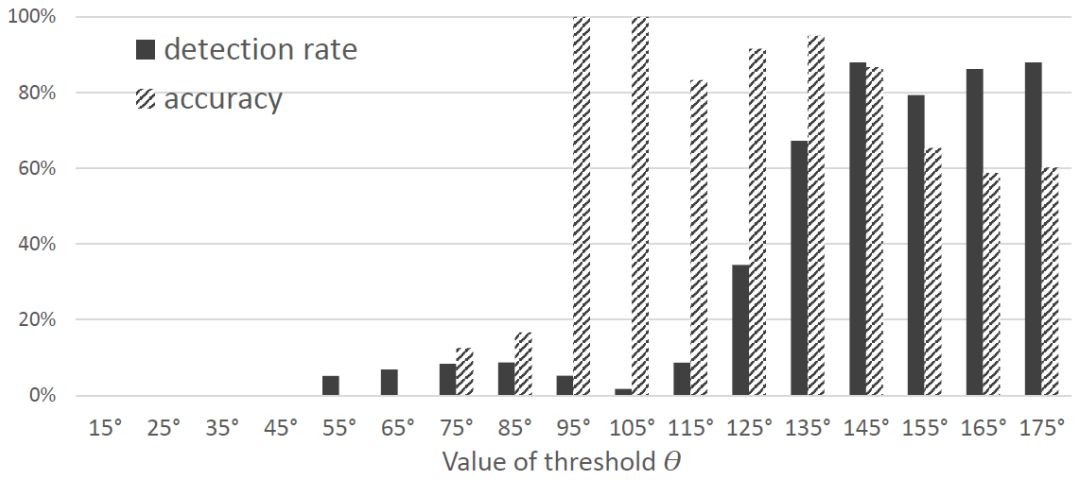


FIGURE 4.4. Detection rate and accuracy of dominant joints using different values of threshold θ.

related to waist and elbows, which are in the range of 16cm and above. Our proposed method demonstrated much-reduced error distance. The error bars in Fig. 4.5 depict the standard deviation (STD) and the average STDs for our proposed method and the SDK are 1.36cm and 0.8cm, respectively. It is evident that the proposed method exhibited much-improved accuracy in comparison to Microsoft Kinect SDK.

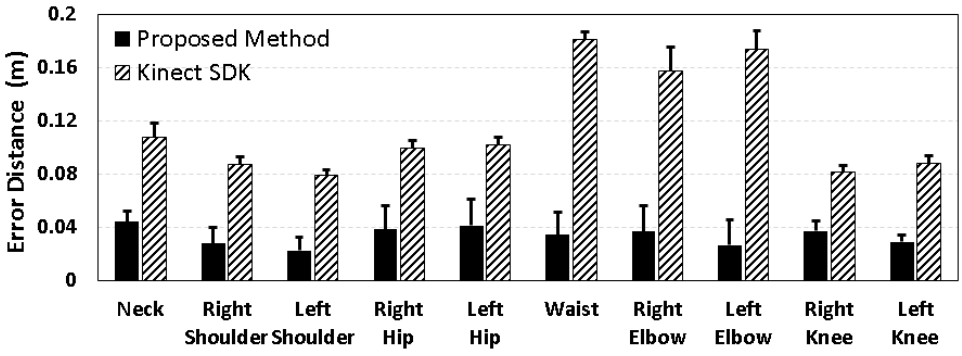


FIGURE 4.5. The average error distance of the detected joints using our proposed method (solid bars) and the Microsoft Kinect SDK (textured bars).

We also evaluated the consistency of joint detection. The consistency is gauged by the distance to the initial detection of each joint. That is, the joint detection of a consistent method deviates slightly, if any, regardless of the poses. Fig. 4.6 illustrates the bar plot of

consistency with respect to the ten joints. Our method exhibited greater consistency for six joints and SDK achieved better consistency for hips, waist, and left knee. The overall average consistencies for our method and the SDK are 3.38cm and 3.8cm, respectively. The error bars in Fig. 4.6 show the standard deviations. The consistencies of the two methods are comparative with a slight advantage to our method.

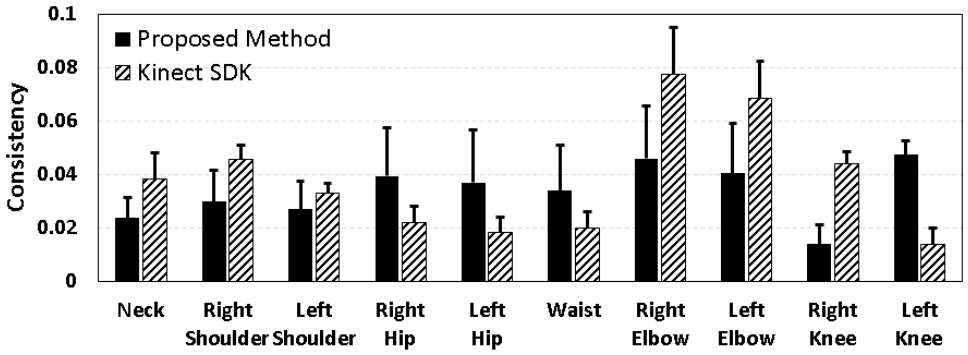
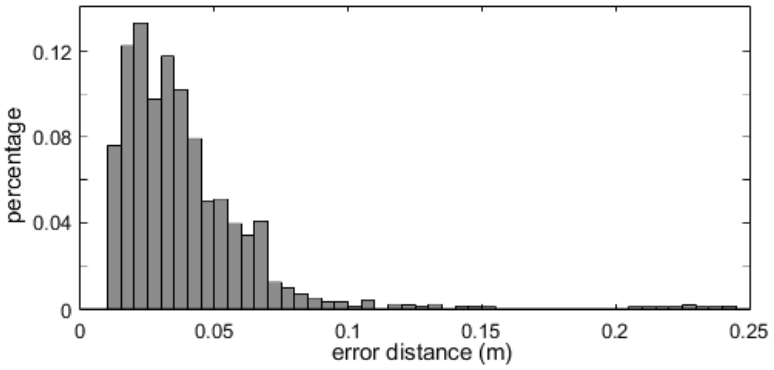


FIGURE 4.6. Average consistency of the detected joints using our proposed method (solid bars) and the Microsoft Kinect SDK (textured bars).

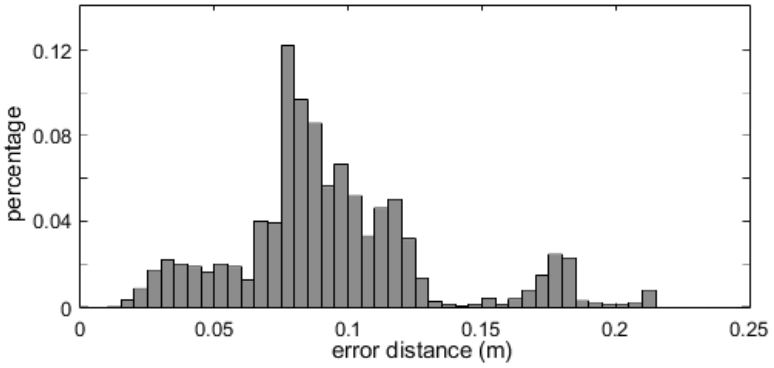
Fig. 4.7 illustrates the histograms of error distance. The distribution of our method is condensed to the lower end and the distribution of the SDK is scattered across the entire scale. The skewness of our method is 1.575 and the skewness of the SDK is 1.091, which indicates that the error distance distribution of our method is statistically better than that of the SDK.

4.2.6. Time Complexity

To evaluate the time complexity of the proposed HJD method, we also implemented the HJD in C++ and tested on a computer with a Intel dual core CPU at 3.4GH and 8GB RAM. The proposed system runs in Windows 8.1 operating system. According to our experiment, the average processing time for each frame was 71.14(ms), which satisfies the need of processing real-time time-of-fight video frame rate.



(a) Our method



(b) Microsoft Kinect SDK

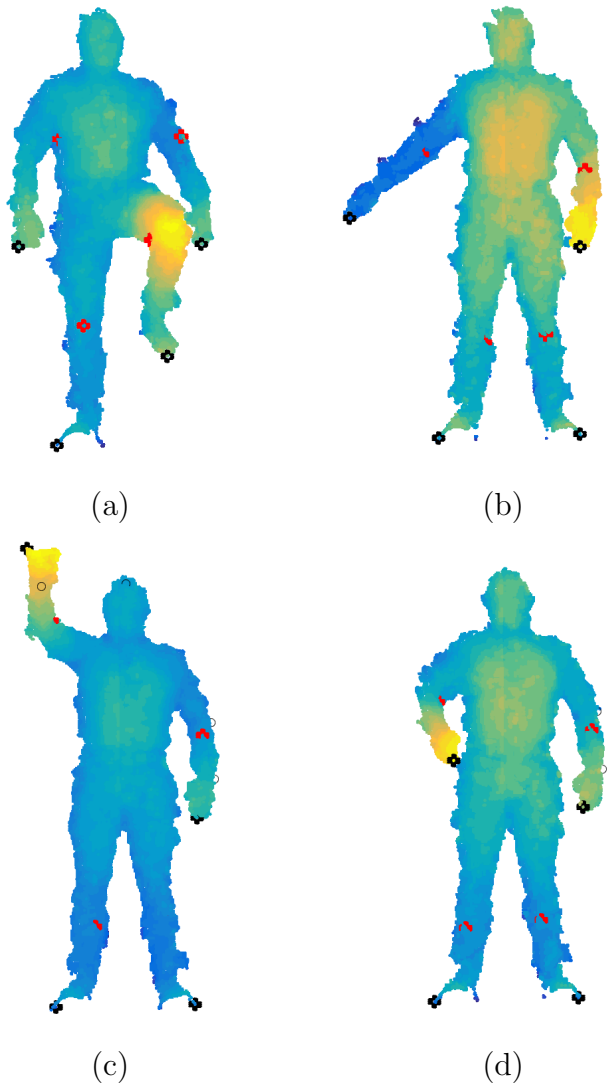
FIGURE 4.7. Histograms of error distance.

4.3. Evaluation for Segmentation based Joint Detection (SJD)

To evaluate the performance of the proposed SJD method, IDT 13 dataset is used. We first evaluate the accuracy of the body part segmentation. Because the results of body part segmentation directly affect the correctness of the joint detection method. Then, we discuss the accuracy and error distance of the joint detection method. Fig. (4.8) shows example results from SJD method.

4.3.1. Error Distance of Body Part Segmentation

Because the joint detection is based on the segmentation results, it is necessary to evaluate the performance of the proposed segmentation method. Firstly, we evaluate the error distance the segmentation results for each limb. The error distance is the Euclidean distance between the middle point of the segmentation boundary and the middle point



(a) (b)

(c) (d)

FIGURE 4.8. Examples of detection results from SJD. The detected joints are marked with red dots. Black dots represent the position of extreme points.

on the reference boundary. Because IDT 13 dataset does not provide the ground truth of the boundaries between different body parts. The reference boundaries used in this evaluation are manually selected by projecting the joint between two body part onto the body surface. Then, all the points with the same geodesic distance to the extreme point on the corresponding limb as the projected joint are selected as the reference boundary. Table 4.6 shows the average error distance for each limb segments.

By observing the results, most of the body part boundaries proposed by the seg-

TABLE 4.6. Error distance of the body part segmentation method

Body part	Left Arm	Right Arm	Left Leg	Right Leg
Error distance(cm)	14.521	12.613	11.426	12.728

mentation method are inside the corresponding limb. This is the main reason which causes the error distance. The proposed segmentation method uses shortest paths on the human body. In some cases, the shortest paths start from the same extreme points start to split into different directions before the all shortest paths reach the end of the limb. Therefore, part of the limb may be left out after the segmentation process. Fig. (4.6) illustrates the gap between proposed boundaries and the reference boundaries.

4.3.2. Error distance of Joint Detection

To evaluate the performance of the joint detection, we first discuss the error distance between the joint detection result and the ground truth from IDT 13 dataset. The error distance is the Euclidean distance between a detected joint and its ground truth.

TABLE 4.7. Error distance of each jointn in SJD

Joint	L Elbow	R Elbow	L Knee	R Knee
Error distance(cm)	18.74	20.23	16.89	19.27

In IDT 13 dataset, the ground truth is recorded by a marker based motion tracking system. The sensors used in IDT 13 are attached to the skin of the human subject. Therefore, the ground truth provided by IDT 13 dataset is not on the surface of the point cloud. The positions of joint detected by SJD are on the surface of the point cloud. Fig. (4.10) shows the position of markers in one frame from different views. Thus, we believe that there is a natural gap even between the ideal detection results and the ground truth.

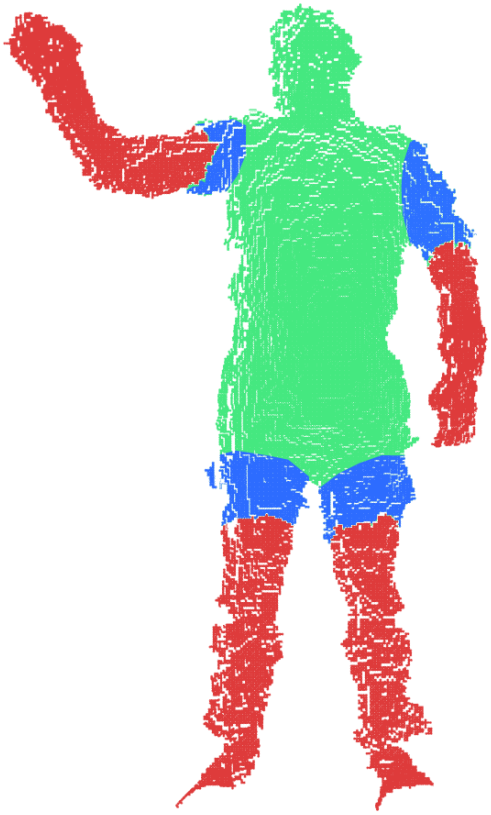
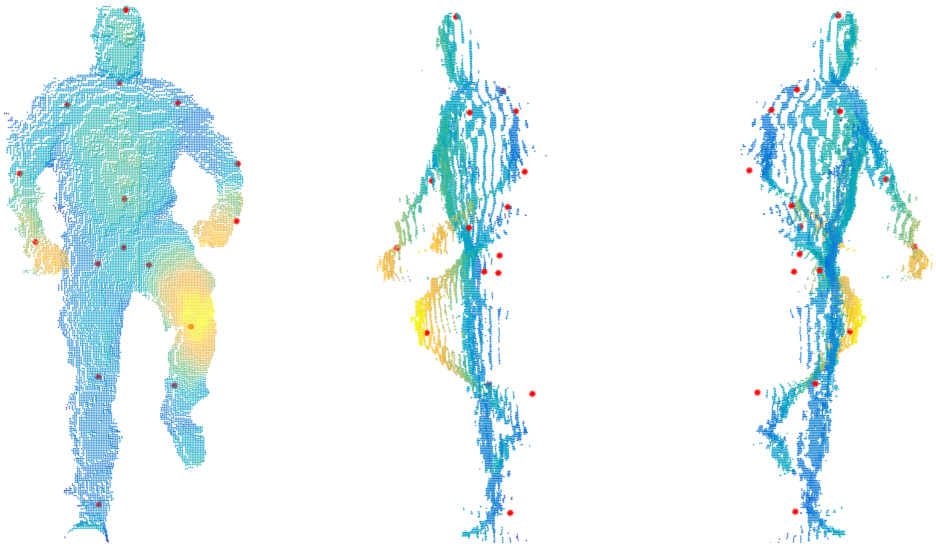


FIGURE 4.9. Illustration for the gap between proposed boundaries and the reference boundaries. The red areas are labeled as limbs, the rest of the human body (both blue and green areas) are labeled as torso. The blue areas are the part left out during the segmentation process and should be labeled as limbs as well.

4.3.3. Accuracy of Joint Detection

The overall average accuracy of the joint detection method is 72.44%. The accuracy of each joint is reported in Table 4.8. Among all the joints, the left and right elbows have higher accuracy than the knees. Because in IDT 13, arms have more activities than the legs. The deformation of arms is also more significant than the deformation on legs. Therefore, elbows have higher accuracy than the knees. Furthermore, the results from body part segmentation directly affect the accuracy of joint detection, because the joint detection method is applied



(a) (b) (c)

FIGURE 4.10. Example of the ground truth of all feature points (including joints). The ground truth of each feature point is labeled by red dots. All ground truth is off the surface. The example is displayed in three views: front view left and right view. (a) shows the front view, (b) shows the left side view, (c) shows the right side view.

on the segmented limbs.

TABLE 4.8. Accuracy of joint detection in SJD

Joint	L Elbow	R Elbow	L Knee	R Knee
Accuracy(%)	74.146	72.16	71.2	72.259

4.4. Failure Cases

Because both HJD and SJD methods use extreme points and shortest paths to detect the joints, we discussed failure cases of both methods together in this section. The experimental results show that both HJD and SJD are capable of detecting joints under the scenarios that self-occlusion occurs and all extreme points are detectable. However, if dif-

ferent body parts are too close (distance between two body parts shorter than 4.5cm), both methods fail to detect the joint on the corresponding limb. Fig. (4.11) shows an example under such scenario. In Fig. (4.11), the left arm of the human subject touches his left leg, the extreme point on the left hand become undetectable because the points on the hand are connected with the points on the torso and the left leg.

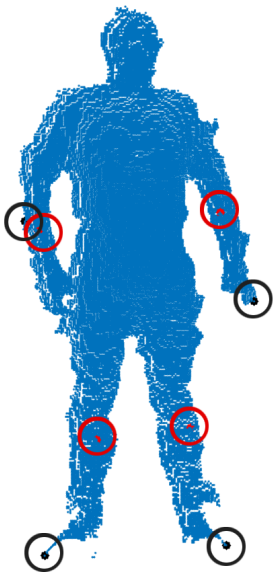
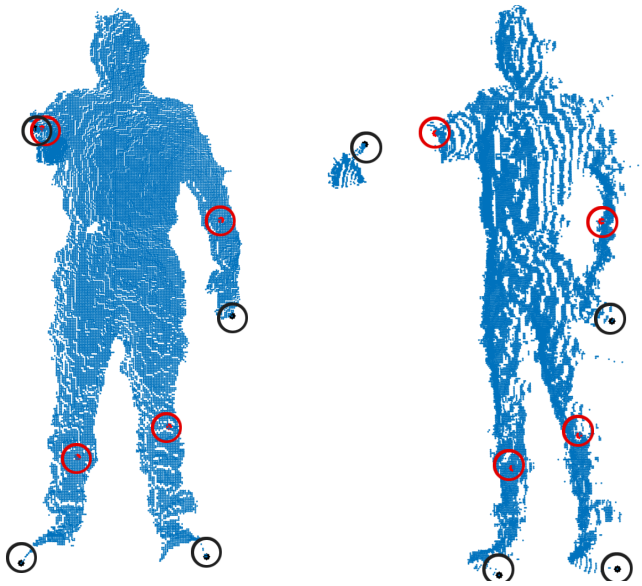


FIGURE 4.11. Example of the failure case with left arm touching the left leg. Red circles indicate the position of joints. Black circles indicate the position of extreme points.

Both methods also fail to detect joints when majority of the data on a limb is missing. Because both methods rely on the local data to detect joints on human body. Fig. (4.12) shows an example result when majority data on the left arm is missing. When a large amount of data is missing on a limb, both methods cannot extract features on the shortest path inside the limb. Thus, the joint detection on such limb gives incorrect results.



(a) (b)

FIGURE 4.12. Example of the failure case with majority data missing on the left arm. (a) shows the front view, (b) shows the right side view.

CHAPTER 5

CONCLUSIONS

In this paper, two joint detection methods with different strategies are proposed. Both methods use local features as the main features to detect joints on the human body. The HJD (hybrid joint detection) relies on the skeleton model to set searching boundaries for each joint. The SJD (segmentation based joint detection) has less dependence on skeleton model than the HJD method. Both methods use the same shortest path extraction method to extract features of the human body. Yet, the SJD relies on the segmentation results to define the boundaries for joint detection instead of skeleton model.

In the HJD method, joints are categorized into two classes, which are implicit joints and dominant joints, according to their deformation degree. The model-based strategy is used to estimate the position of implicit joints. A loose skeleton model is introduced in the model-based strategy. Unlike other rigid 3D skeletons, the skeleton in HJD is a set of rules, which constrain the searching area of each joint, based on the geodesic features of the human body. The advantage of using such skeleton model is that the more flexibility is provided for locating joints. For implicit joints, an optimization procedure is applied to find the optimized position of them. For dominant joints, the skeleton only defined the boundaries for searching them, so that the joint detection method can provide more accurate results. The data-driven strategy is used to detect the dominant joints. Both strategies take advantage of the geodesic features of the human body to accurately locate the joints.

The SJD method mainly use local data to detect joints. In SJD method, a human body is divided into torso and limbs by the proposed human body segmentation method. The proposed segmentation method takes advantage of the features extracted from shortest paths on a human body to find the boundaries between limbs and the torso. Then, the joints are detected from the segmented limbs. Skeleton model is also used in SJD. Unlike the HJD, SJD does not use skeleton model to define the searching boundaries. The skeleton model provides the geodesic features of a human body so that the detected joints can be assigned

to the correct semantic labels. A likelihood function is proposed to detect the joints inside limbs.

Both HJD and SJD methods extract features from local data and leverage the geodesic features of the human body to detect joints. The contribution of HJD is that a hybrid joint detection framework, which combines data-driven and model-based strategies, is proposed. Such framework mainly relies on local data instead of model fitting procedure. Therefore, it can be used for time sensitive applications potentially. The contribution of SJD is that a data-driven human body segmentation method and a joint likelihood function is proposed. The proposed segmentation method can also be used for other applications. Our experimental results demonstrated that both methods provide accurate and robust results. Furthermore, the data-driven method that uses global shortest path and local shortest path can be widely used in different types of methods for human pose detection. The geodesic distances between the extreme points and the joints can be used for tracking and estimating the position of joints when the joints are occluded. However, there are a few issues exposed during the experiments. Complex and multi-layer self-occlusions could cause failure of detection in both methods. Both methods failed to detect the joints when the body parts and limbs are occluded. A large amount of missing data on limbs also cause incorrect detection in both methods. This situation usually happens when the human subject points his arm or leg directly to the camera.

In our future work, we plan to employ the temporal information between video frames to improve the accuracy and robustness of the detection for both methods. For the HJD method, detection of implicit joints needs to be improved. When detecting dominant joints, more features will be explored based on the local data and shortest path to improve the overall accuracy of dominant joints. For SJD method, current segmentation strategy needs to be improved to provide more stable segmentation results. Current segmentation strategy relys on fixed threshold, therefore an adaptive threshold selection method can potentially stablize the the segmentation results in SJD. Shortest path extraction also needs to be improved to reduce the affection caused by the deformation of clothes and human body.

Extreme points detection can be improved by applying tracking method or optical flow to correctly detect the extreme point when it is connected to other body parts. In addition, to solve the issue caused by missing data, more model-based strategies can be used in both systems.

REFERENCES

[1] A. Abobakr, M. Hossny, and S. Nahavandi, *Body joints regression using deep convolutional neural networks*, 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct 2016, pp. 3281–3287.

[2] A. Baak, M. Muller, G. Bharaj, H.P. Seidel, and C. Theobalt, *A data-driven approach for real-time full body pose reconstruction from a depth camera*, IEEE International Conference on Computer Vision (Barcelona), Nov. 6-13 2011, pp. 1092–1099.

[3] M. D. Bengalur, *Human activity recognition using body pose features and support vector machine*, 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Aug 2013, pp. 1970–1975.

[4] Z. P. Bian, J. Hou, L. P. Chau, and N. Magnenat-Thalmann, *Fall detection based on body part tracking using a depth camera*, IEEE Journal of Biomedical and Health Informatics 19 (2015), no. 2, 430–439.

[5] Koen Buys, Cedric Cagniart, Anatoly Baksheev, Tinne De Laet, Joris De Schutter, and Caroline Pantofaru, *An adaptable system for rgb-d based human body detection and pose estimation*, Journal of Visual Communication and Image Representation 25 (2014), no. 1, 39 – 52.

[6] Z. Cai, J. Han, L. Liu, and L. Shao, *Rgb-d datasets using microsoft kinect or similar sensors: a survey*, Multimedia Tools and Applications 76 (2017), no. 3, 4313–4355.

[7] A. A. Chaaraoui, J. R. Padilla-López, P. Climent-Pérez, and F. Flórez-Revuelta, *Evolutionary joint selection to improve human action recognition with rgb-d devices*, Expert Systems with Applications 41 (2014), no. 3, 786 – 794.

[8] A. A. Chaaraoui, J. R. Padilla-López, and F. Flórez-Revuelta, *Fusion of skeletal and silhouette-based features for human action recognition with rgb-d devices*, 2013 IEEE International Conference on Computer Vision Workshops, Dec 2013, pp. 91–97.

[9] S. Chandra, S. Tsogkas, and I. Kokkinos, *Accurate human-limb segmentation in rgb-d*

images for intelligent mobility assistance robots, 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), Dec 2015, pp. 436–442.

[10] Siyuan Chen, F. Bremond, Hung Nguyen, and H. Thomas, *Exploring depth information for head detection with depth images*, 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Aug 2016, pp. 228–234.

[11] S. Y. Chun and C. S. Lee, *Applications of human motion tracking: Smart lighting control*, 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops, June 2013, pp. 387–392.

[12] J. Cui, Y. Liu, Y. Xu, H. Zhao, and H. Zha, *Tracking generic human motion via fusion of low- and high-dimensional approaches*, IEEE Transactions on Systems, Man, and Cybernetics: Systems 43 (2013), no. 4, 996–1002.

[13] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, *Real time motion capture using a single time-of-flight camera*, IEEE Conference on Computer Vision and Pattern Recognition (2010), 755 – 762.

[14] K. Greff, A. Brando, S. Krau, D. Stricker, and E. Clua, *A comparison between background subtraction algorithms using a consumer depth camera*, International Conference on Computer Vision Theory and Applications (2012), 431–436.

[15] S. Handrich and A. Al-Hamadi, *A robust method for human pose estimation based on geodesic distance features*, IEEE International Conference on Systems, Man, and Cybernetics, 2013, pp. 906 – 911.

[16] T. Helten, M. Müller, H. P. Seidel, and C. Theobalt, *Real-time body tracking with one depth camera and inertial sensors*, 2013 IEEE International Conference on Computer Vision, Dec 2013, pp. 1105–1112.

[17] A. Jalal and Y. Kim, *Dense depth maps-based human pose tracking and recognition in dynamic scenes using ridge data*, IEEE International Conference on Advanced Video and Signal Based Surveillance, 2014, pp. 119–124.

[18] S. R. Ke, L. Zhu, J. N. Hwang, H. I. Pai, K. M. Lan, and C. P. Liao, *Real-time 3d human pose estimation from monocular view with applications to event detection and*

video gaming, 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, Aug 2010, pp. 489–496.

[19] L.-C.Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, *Semantic image segmentation with deep convolutional nets and fully connected crfs*, (2014).

[20] Mun Wai Lee and I. Cohen, *A model-based approach for estimating human 3d poses in static images*, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006), no. 6, 905–916.

[21] S. Li, W. Zhang, and A. B. Chan, *Maximum-margin structured learning with deep networks for 3d human pose estimation*, 2015 IEEE International Conference on Computer Vision (ICCV), Dec 2015, pp. 2848–2856.

[22] Z. Li and D. Kulic, *Local shape context based real-time endpoint body part detection and identification from depth images*, 2011 Canadian Conference on Computer and Robot Vision, May 2011, pp. 219–226.

[23] X. Ma, H. Wang, B. Xue, M. Zhou, B. Ji, and Y. Li, *Depth-based human fall detection via shape features and improved extreme learning machine*, IEEE Journal of Biomedical and Health Informatics 18 (2014), no. 6, 1915–1922.

[24] T. B. Moeslund, A. Hilton, and V. Krüger, *A survey of advances in vision-based human motion capture and analysis*, Computer Vision and Image Understanding 104 (2006), no. 2, 90 – 126.

[25] T. B. Moeslund, A. Hilton, V. Krüger, and Eds L.Sigal, *Visual analysis of humans—looking at people*, Springer, New York, NY, USA, 2011.

[26] R. Pinho and J. Tavares, *Tracking features in image sequences with kalman filtering, global optimization, mahalanobis distance and a management model*, Computer Modeling in Engineering & Sciences 46 (2009), no. 1, 51–75.

[27] R. Pinho, J. Tavares, and M. Correia, *A movement tracking management model with kalman filtering, global optimization techniques and mahalanobis distance*, Lecture Series on Computer and Computational Sciences 4A (2005), 463–466.

[28] ———, *An improved management model for tracking missing features in computer vi-*

sion long image sequences, WSEAS Transactions on Information Science and Applications 1 (2007), no. 4, 196–203.

[29] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, *Real-time identification and localization of body parts from depth images*, IEEE International Conference on Robotics and Automation (2010), 3108 – 3113.

[30] L.A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, *Estimating human 3D pose from time-of-flight images based on geodesic distances and optical flow*, IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (2013), 700 – 706.

[31] J. Shotton, R. Girshick, A. Fitzgibbon, T.Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, and and A. Blake A. Kipman, *Efficient human pose estimation from single depth images*, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2013), no. 12, 2821 – 2840.

[32] M. Sigalas, M. Pateraki, and P. Trahanias, *Full-body pose tracking-the top view reprojection approach*, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (2016), no. 8, 1569–1582.

[33] Jaeyong Sung, C. Ponce, B. Selman, and A. Saxena, *Unstructured human activity detection from rgbd images*, 2012 IEEE International Conference on Robotics and Automation, May 2012, pp. 842–849.

[34] J. Tavares and A. Padilha, *Matching lines in image sequences using geometric constraints*, RecPad'95-7th Portuguese Conference on Pattern Recognition (Portugal), 1995.

[35] M. Vasconcelos and J. Tavares, *Human motion segementation using active shape models*, pp. 237–246, 2015.

[36] Q. Wang, G. Kurillo, F. Ofli, and R. Bajcsy, *Evaluation of pose tracking accuracy in the first and second generations of microsoft kinect*, IEEE International Conference on Healthcare Informatics (Dallas, TX), 2015.

[37] X. Wei, P. Zhang, and J. Chai, *Accurate realtime full-body motion capture using a single*

depth camera, ACM Transactions on Graphics - Proceedings of ACM SIGGRAPH Asia Volume 31 Issue 6 (2012), Article No. 188.

[38] E.J. Weng and L.C. Fu, *On-line human action recognition by combining joint tracking and key pose recognition*, IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 7-Oct. 12 2012, pp. 4112 – 4117.

[39] Y. Xiao, P. Siebert, and N. Werghi, *Topological segmentation of discrete human body shapes in various postures based on geodesic distance*, Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., vol. 3, Aug 2004, pp. 131–135 Vol.3.

[40] Hee-Deok Yang and Seong-Whan Lee, *Reconstructing 3d human body pose from stereo image sequences using hierarchical human body model learning*, 18th International Conference on Pattern Recognition (ICPR'06), vol. 3, 2006, pp. 1004–1007.

[41] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, *Accurate 3d pose estimation from a single depth image*, 2011 International conference on Computer Vision (Barcelona), 2011, pp. 731–738.

[42] X. Yuan, L. Kong, D. Feng, and Z. Wei, *Automatic feature point detection and tracking of human actions in time-of-flight videos*, IEEE/CAA Journal of Automatica Sinica In press (2017).

[43] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, and C. Tang, *Rgb-d-based action recognition datasets: A survey*, Pattern Recognition 60 (2016), no. Supplement C, 86 – 105.

[44] X. Zhang, C. Li, W. Hu, X. Tong, S. Maybank, and Y. Zhang, *Human pose estimation and tracking via parsing a tree structure based human model*, IEEE Transactions on Systems, Man, and Cybernetics: Systems 44 (2014), no. 5, 580–592.

[45] Y. Zhu, B. Dariush, and K. Fujimura, *Controlled human pose estimation from depth image streams*, 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, June 2008, pp. 1–8.

[46] ———, *Kinematic self retargeting: A frameworks for human pose estimation*, Computer Vision and Image Understanding 114 (2010), no. 12, 1362 – 1375.

[47] S. Zuffi and M. Black, *The stitched puppet: A graphical model of 3d human shape and pose*, IEEE Conference on Computer Vision and Pattern Recognition, June 2015, pp. 3537–3546.